UNIVERSITÀ DI PISA

Corso di Laurea Magistrale in Informatica Umanistica

TESI DI LAUREA MAGISTRALE

Aspetti linguistici e gradimento degli utenti negli app store

Candidato: *Sara Perrone*

Relatore: *Prof. Vincenzo Gervasi*

Anno Accademico 2014-2015

Indice

-1-

Introduzione

1.1 Obiettivo della tesi

Il presente lavoro si pone l'obiettivo di analizzare le recensioni prodotte dagli utenti all'interno degli app store. L'analisi si svolge attraverso lo studio di aspetti di carattere linguistico poco affrontati nella ricerca, in relazione alla valutazione espressa dagli utenti.

Gli app store, sono piattaforme online dove è possibile scaricare e pubblicare applicazioni mobili. Essi costituiscono un vero e proprio punto di incontro tra produttore e consumatore, che sono messi in contatto in maniera diretta: il produttore può presentare il suo prodotto tramite una descrizione e il consumatore può commentare e valutare il prodotto o anche chiedere assistenza.

Un aspetto importante delle descrizioni delle applicazioni è che sono il primo strumento con cui gli sviluppatori comunicano con gli utenti. Risulta dunque interessante cercare di capire quale è il rapporto dell'utente nei confronti delle descrizioni, ad esempio se vengono lette in parte o totalmente, se considerano solo le immagini, ecc.

Vari studi proposti in letteratura hanno sino ad ora trascurato un aspetto che forse merita una più attenta valutazione se si pensa che la prima fonte di informazioni su un'applicazione è sicuramente la descrizione. La prima parte dell'indagine si focalizza inizialmente sul testo presente nelle descrizioni con l'obiettivo di scoprire se esistono legami tra ciò che viene espresso nella descrizione e l'opinione che gli utenti hanno dell'applicazione.

La strategia utilizzata prevede di individuare un numero di parole chiave capaci di rappresentare gli aspetti importanti di una descrizione. Ricercando queste parole chiave all'interno delle recensioni degli utenti si può verificare se esistono relazioni

tra la presenza di queste parole e la valutazione espressa dall'utente.

Un secondo aspetto ritenuto interessante, è il modo in cui gli utenti utilizzano la scala di valori proposta negli app store, la quale permette di assegnare un voto di gradimento ad una applicazione. Tale scala prevede cinque valori (da 1 a 5) che danno la possibilità di esprimere, oltre a un valore binario positivo-negativo, anche delle sfumature comprese tra questi valori. Queste sfumature, composte dai valori intermedi (2, 3, 4), non sono definite all'utente in ciò che dovrebbero rappresentare, lasciando a quest'ultimo libera interpretazione. La seconda parte dell'indagine mira a determinare, attraverso lo studio delle recensioni, se la scala di valori soddisfi gli utenti nell'atto di esprimere la loro opinione.

Generalmente è possibile affermare che un testo presenta caratteristiche linguistiche (morfologiche, sintattiche, semantiche, ecc.) che dipendono strettamente dall'informazione che trasmette e dal contesto in cui esso viene prodotto.

La strategia, dunque, consiste nell'individuazione di caratteristiche linguistiche nel testo delle recensioni, che indichino un sentimento positivo o negativo, in modo da riuscire a definire se le valutazioni assegnate dagli utenti siano coordinate con la positività o la negatività espressa dal testo che le accompagna.

Questa tesi di carattere esplorativo esamina più di 200 applicazioni presenti in Google Play Store di Android, attraverso l'uso di tecniche di analisi automatica del testo, ampiamente studiate e ormai consolidate.

1.2 L' importanza delle applicazioni

Negli ultimi 5 anni si è assistito ad una significativa rivoluzione nel mercato del mobile grazie all'avvento di nuovi dispositivi come smartphone e tablet.

Tale rivoluzione è dovuta alla nascita delle applicazioni mobili e in particolare alla nascita del primo App Store, quello di Apple, che aveva lo scopo di creare un nuovo modo di distribuzione di software, cioè un “negozio online” dove facilmente si potevano acquistare o pubblicare applicazioni di qualsiasi genere.

Nell'arco di pochi mesi anche Google ha lanciato il proprio negozio virtuale “Android Market”, conosciuto oggi come Play Store.

Inizialmente le applicazioni disponibili non erano tante, ma nel giro di pochi mesi il loro numero è cresciuto notevolmente incontrando subito l'interesse della gente e, considerato che le regole per creare un app non erano restrittive, molti sono entrati a far parte di questo mondo creando la propria applicazione.

I vari App Store permettono a grandi e piccole aziende come anche a sviluppatori indipendenti di pubblicare e vendere la propria applicazione, ciò determina un continuo aumento delle applicazioni disponibili.

Oggi il numero di applicazioni di Play Store, secondo le stime di AppBrain[1], supera i 2 milioni, mentre il numero di applicazioni di Apple App Store, secondo AppShopper[2], supera di poco un 1.700.000 .

L'Osservatorio Mobile Economy del Politecnico di Milano[3], che studia l'evoluzione dei mercati dell'ecosistema Mobile, durante il convegno di CorCom “Telco per l’Italia ”[4] tenuto a Roma il 21 maggio del 2015, ha presentato i dati dell'impatto della Mobile Economy in Italia, ovvero quella porzione di economia relativa al mondo Mobile.

Secondo le stime relative al 2015 quest'ambito vale 25,7 miliardi di euro, cioè 1,65% del PIL italiano, con previsione per il 2017, come si può vedere in Figura 1, di 37 miliardi di euro pari al 2,3% del PIL, grazie soprattutto al crescente utilizzo di servizi come Mobile Commerce e Mobile Payement.

1 http://www.appbrain.com/stats/number-of-android-apps
2 http://appshopper.com/
3 http://www.osservatori.net/mobile-economy
4 http://www.corrierecomunicazioni.it/eventi/33757_telco-per-l-italia.htm

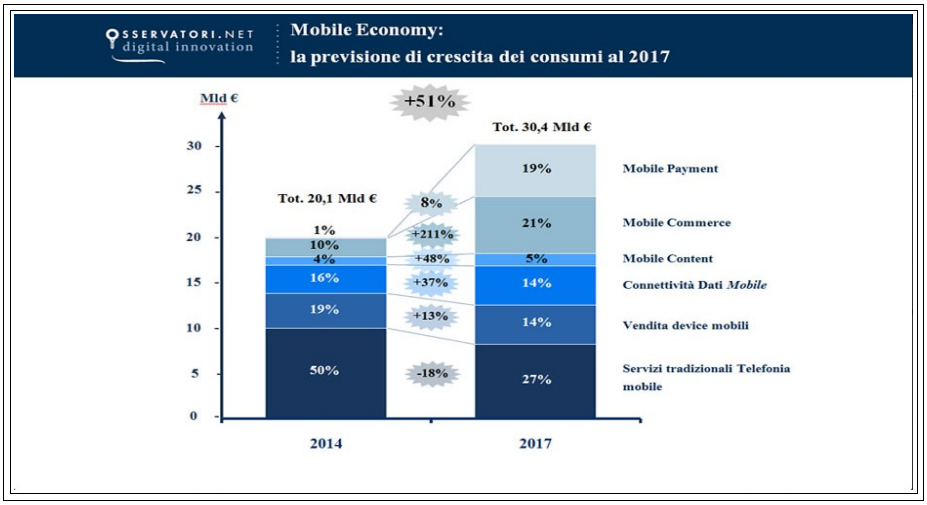


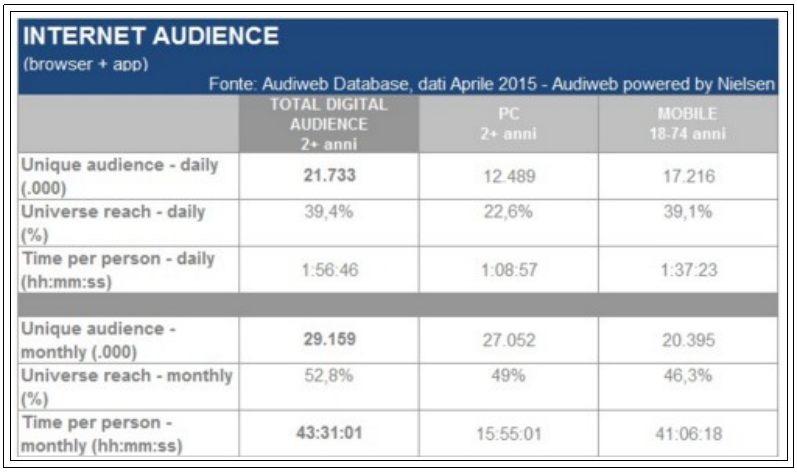
Figura 1. Dati Osservatorio Mobile Economy del Politecnico di Milano.

Tutto questo fa capire che il mercato delle applicazioni mobili è diventato decisamente appetibile per le aziende, anche in Italia, soprattutto perché è un mercato in continua evoluzione con un numero di utenti sempre crescente.

Al giorno d'oggi sempre più persone utilizzano i dispositivi mobili per connettersi ad internet, tanto che il tempo trascorso dagli utenti a navigare sul web non è legato al classico computer ma ai dispositivi mobili come lo smartphone, considerato oramai un elemento insostituibile.

Le connessioni da smartphone e tablet, secondo i dati pubblicati da Audiweb[5] (società che rileva e distribuisce i dati di audience di internet in Italia) relativi già al mese di aprile 2015, hanno soppiantato quelle da PC.

5 http://www.audiweb.it/news/total-digital-audience-del-mese-di-aprile-2015/



INTERNET AUDIENCE
(browser + app)
Fonte: Audiweb Database, dati Aprile 2015 - Audiweb powered by Nielsen

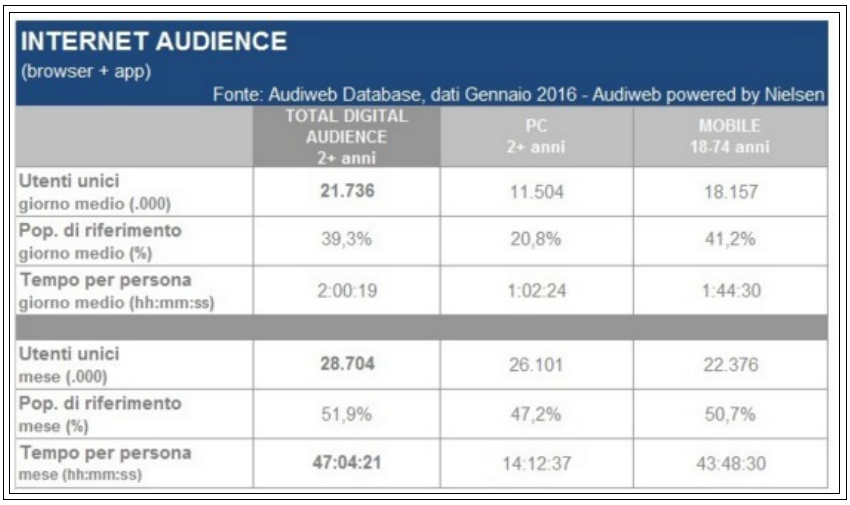
	TOTAL DIGITAL AUDIENCE 2+ anni	PC 2+ anni	MOBILE 18-74 anni
Unique audience - daily (.000)	21.733	12.489	17.216
Universe reach - daily (%)	39,4%	22,6%	39,1%
Time per person - daily (hh:mm:ss)	1:56:46	1:08:57	1:37:23
Unique audience - monthly (.000)	29.159	27.052	20.395
Universe reach - monthly (%)	52,8%	49%	46,3%
Time per person - monthly (hh:mm:ss)	43:31:01	15:55:01	41:06:18

Figura 2. Dati Audiweb Aprile 2015.

Come mostrato in figura 2 gli accessi unici giornalieri a internet sono stati in media 21.733 milioni. Di questi 12.489 milioni sono stati effettuati da PC e 17.216 milioni da dispositivi mobili. Gli utenti monitorati rientrano nell'età da 2 anni in su per quanto riguarda l'audience da PC, e da 18 a 74 anni per l'audience da dispositivi mobili.

Negli ultimi dati[6] pubblicati da Audiweb relativi al mese di gennaio 2016 l'audience da dispositivi mobile giornalmente è aumentato di quasi un milione raggiungendo quota 18.157 milioni di utenti, il 41.2% degli italiani collegati da qualsiasi dispositivo mobile (figura 3).

6 http://www.audiweb.it/news/total-digital-audience-del-mese-di-gennaio-2016/



INTERNET AUDIENCE
(browser + app)
Fonte: Audiweb Database, dati Gennaio 2016 - Audiweb powered by Nielsen

	TOTAL DIGITAL AUDIENCE 2+ anni	PC 2+ anni	MOBILE 18-74 anni
Utenti unici giorno medio (.000)	**21.736**	11.504	18.157
Pop. di riferimento giorno medio (%)	39,3%	20,8%	41,2%
Tempo per persona giorno medio (hh:mm:ss)	2:00:19	1:02:24	1:44:30
Utenti unici mese (.000)	**28.704**	26.101	22.376
Pop. di riferimento mese (%)	51,9%	47,2%	50,7%
Tempo per persona mese (hh:mm:ss)	**47:04:21**	14:12:37	43:48:30

Figura 3. Dati Audiweb Gennaio 2016.

I dispositivi mobili rappresentano sempre più lo strumento a cui gli italiani ricorrono per la fruizione di internet, come si può dedurre dall'indagine condotta da Audiweb.

Il continuo espandersi di questo mercato intorno alle applicazioni mobili suscita interesse anche da parte del mondo della ricerca, come evidenziato dalla presenza di studi effettuatati in questo ambito (D. Pagano, W. Maalej, 2013). Tale interesse è dovuto anche alla quantità di dati generati attraverso le recensioni degli utenti.

Questo lavoro si basa esclusivamente su dati recuperati da Google Play Store, sebbene esistano anche altri app store.

Le applicazioni presenti in questo store sono suddivise per categoria che rispecchia l'ambito del servizio che offrono, ad esempio applicazioni per la messaggistica, per la gestione di spese, applicazioni per l'editing di immagini e foto, ecc.

Di ogni applicazione è possibile visualizzare la descrizione e il voto in “stelle” ottenuto dalla media del totale delle valutazioni degli utenti; tale voto è un numero compreso tra uno e cinque e incide sulla popolarità di un'applicazione. Questo è molto importante perché influisce sull'ordine con cui le applicazioni vengono mostrate agli utenti.

Oltre alla descrizione, l'utente ha la possibilità di vedere anche le recensioni espresse

dagli altri utenti, che presentano un voto anch'esso in stelle che va da uno a cinque.

Le recensioni insieme alle valutazioni giocano un ruolo rilevante nel determinare la posizione in classifica dell'applicazione, che serve a sua volta per avere maggiore visibilità e ottenere quindi maggiori download.

Le recensioni contengono vari generi di informazione, malfunzionamenti dell'applicazione, consigli utili, semplici giudizi positivi o negativi. Esse forniscono un feedback prezioso per lo sviluppatore ai fini di migliorare il proprio prodotto e valutare la soddisfazione degli utenti.

Per questo risulta vantaggioso per chi produce applicazioni sfruttare l'informazione espressa dagli utenti per avere maggiore successo, anche se non sempre la grossa quantità di recensioni permette un'agevole lettura delle stesse. Si rende dunque necessario avere strumenti e metodi che permettano di facilitare il trattamento di tali dati. Un esempio può essere quello di raggruppare e filtrare le recensioni in base alla loro utilità.

Il ruolo che la ricerca ha svolto e continua a svolgere è fornire approfondimenti utili per i produttori di questi strumenti e per gli sviluppatori di applicazioni, oltre che fornire indicazioni su quali informazioni raccogliere e come recuperarle.

1.3 Struttura tesi

Nella prima parte della tesi si dà una panoramica delle discipline coinvolte, cosa sono, quali tecniche e strumenti utilizzano quando sottopongono ad analisi grosse quantità di dati e l'ambito in cui operano.

Si pone l'attenzione sull'analisi automatica di dati di tipo testuale, in particolare quando il testo analizzato è un testo non strutturato, che è il caso delle recensioni di applicazioni mobili (Sezione 2).

Nella seconda parte vengono presentati i quesiti alla base dello studio sperimentale, e la metodologia seguita nel tentativo di rispondere a tali domande. Infine è riportata una breve descrizione per tutti gli strumenti che sono stati utilizzati nel corso del

lavoro (Sezione 3).

Nella sezione successiva sono state descritte, da un punto di vista tecnico, le fasi svolte per raccogliere i dati grezzi, che costituiscono due corpora, punto di partenza del lavoro. A seguire è stata riportata la fase di estrazione dai corpora dei dati necessari agli esperimenti di analisi (Sezione 4).

La sezione 5 presenta l'intera analisi svolta, effettuata formalizzando ipotesi con lo scopo di rispondere ai quesiti posti in questo lavoro. Di ogni ipotesi sono state riportati esperimenti e relativi risultati ottenuti (Sezione 5).

La sezione seguente consiste nell'interpretazione e discussione dei risultati esposti nella sezione precedente in relazione ai quesiti posti (Sezione 6).

L'ultima sezione della tesi è dedicata alle considerazioni finali sullo studio svolto (Sezione 7).

-2-

Analisi automatica dei testi

Grandi quantità di dati vengono generati da ciò che ci circonda in ogni momento. Ogni processo digitale produce informazione potenzialmente utile che viene registrata nelle banche dati in tutto il mondo. Questa vasta mole di dati rappresenta un importante patrimonio informativo che, se sfruttato, può portare alla luce nuova conoscenza utile e vantaggiosa.

I dati provengono dalle fonti più disparate (acquisti online, dati GPS, tabulati telefonici, interazioni con social media, ecc.) e sono comprensibilmente di natura diversa. Tutti questi dati possono essere suddivisi in tre macro-categorie: dati strutturati, non strutturati e semi-strutturati. Nel primo gruppo rientrano tutti quei dati che si rifanno a uno schema rigido, come record di database relazionali, nel secondo i dati che non hanno alcuna struttura esplicita, come ad esempio un testo generico; e infine i dati semi-strutturati sono quelli che presentano strutture variabili che dipendono dai dati stessi, l'esempio più immediato sono i file XML.

Il potenziale informativo che i dati offrono dipende dalla capacità di estrarre e analizzare l'informazione dai dati stessi. Sono varie le tecniche e le metodologie usate a questo scopo, la cui scelta è determinata dal tipo di dato studiato.

Le tre categorie strutturali di dati si differenziano anche per numerosità, questo è noto nel campo della Business Intelligence dove è stato stimato che l'80-85% dei *business data* sono dati non strutturati (Blumberg, R., & Atre, S, 2003; S. Grimes, 2008).

Immagini, video e testo libero rientrano in questa tipologia di dato e sono quei dati che vengono generati consapevolmente dagli utenti sul web attraverso *social network* o più generalmente attraverso piattaforme di condivisione. Ad oggi questi dati, in particolare quelli testuali, ispirano campagne di marketing di aziende, le quali sono sempre più interessate a scoprire cosa pensano e cosa vogliono i loro clienti.

L'estrazione di informazioni significative da questi dati non strutturati può essere fondamentale per il successo di un'azienda.

In questa tesi vengono trattati solo i dati non strutturati di tipo testuale.

Le tecniche che si occupano di estrarre e analizzare informazione da un testo rientrano nell'ambito dell'analisi automatica del testo (o t*ext analytics*).

L'analisi automatica del testo è quell'insieme di tecniche che consentono di trattare grandi quantità di dati testuali, con lo scopo di ricavare conoscenza implicita che altrimenti rimarrebbe nascosta e non sfruttata.

Interagire con enormi masse di materiali testuali, spesso disponibili in rete, porta il problema di selezionare, all'interno di questa fonte smisurata, i dati di interesse per estrarne informazione capace di produrre valore.

Si tratta di un campo multidisciplinare che per raggiungere questo obiettivo ha bisogno di coinvolgere l'*information retrieval*, la linguistica computazionale, il *natural language processing*, la statistica e il *machine learning*. La collaborazione tra queste discipline consente di portare alla luce nuova conoscenza.

L'analisi automatica del testo trova applicazione in qualunque campo di indagine che richiede per il proprio fine un trattamento di grandi quantità di dati testuali, come per esempio *business intelligence*, *email filtering*, ecc.

Il processo di analisi automatica del testo, come si può vedere in figura 4: si può suddividere in 4 fasi principali:

- recupero dei testi;
- estrazione dell'informazione da essi;
- analisi dei dati ottenuti;
- interpretazione dei risultati.

Figura 4. Fasi analisi automatica del testo.

Il primo passo consiste nel recuperare e raggruppare i testi oggetto di studio, che devono essere adatti al tipo di analisi che si ha intenzione di svolgere. Può essere necessario in questa fase l'uso dell'*information retrieval*, che svolge processi di suddivisione e di selezione di documenti o testi rilevanti a partire da un insieme eterogeneo.

La seconda fase riguarda l'*information extraction*, il cui obiettivo è quello di prelevare informazioni da un documento testuale non strutturato e riportarle in una forma strutturata attraverso l'uso di teorie di linguistica computazionale e tecniche di natural language processing. Questo processo, conosciuto anche come *feature extraction*, è necessario per fornire rappresentazioni del contenuto dei testi oggetto di studio e per estrarre da questi alcune proprietà, attraverso misurazioni di tipo quantitativo; in sostanza consiste nel trasformare il testo in dati numerici adatti all'analisi.

La trasformazione di proprietà del testo o delle sue componenti è strettamente legata all'obiettivo prefissato. È difficile dunque definire a priori l'insieme di proprietà e caratteristiche potenzialmente utilizzabili del testo, data la varietà di applicazioni e impieghi dell'analisi dei testi. Una distinzione grossolana che può essere fatta è suddividere le proprietà in base ad aspetti di tipo linguistico o meno. Ciò non toglie che un qualunque studio di analisi automatica del testo possa basarsi su una combinazione di entrambe le tipologie, strategia che ne può rafforzare l'efficacia.

Nella terza fase, una volta ottenuti i dati nella loro forma strutturata, vengono analizzati sfruttando applicazioni di algoritmi di machine learning, i quali permettono di costruire modelli statistici imparando iterativamente dai dati stessi.

Il vantaggio più importante di queste tecniche di indagine statistica è la loro capacità di evidenziare relazioni o pattern tra i dati senza che siano esplicitamente programmate a farlo.

Infine l'ultima fase consiste nell'interpretazione dei risultati, anch'essa strettamente dipendente dal tipo di analisi e dal genere di risultati stessi.

Nel presente lavoro sono state utilizzate tecniche consolidate di linguistica computazionale, *natural language processing* e *sentiment analysis*, seguite da metodologie di predizione appartenenti al *machine learning*.

Nei prossimi capitoli sono illustrate queste discipline.

2.1 Linguistica computazionale

La linguistica computazionale, è un settore interdisciplinare che si occupa di studiare il linguaggio naturale, o linguaggio umano, mediante l'uso di strumenti informatici. Si concentra sull'analisi di dati linguistici sfruttando metodi matematici e statistici e tecniche informatiche per sviluppare modelli computazionali della lingua.

La linguistica computazionale si pone l'obiettivo di rendere il computer capace di comprendere la struttura e il contenuto dei testi e di acquisire le competenze necessarie per trattare il linguaggio naturale come un essere umano. Tuttavia, allo

stato attuale gli strumenti più sofisticati che si hanno a disposizione sono lontani dal poter soddisfare queste aspettative.

Le origini di questa disciplina si basano su due diversi filoni di ricerca, il primo è rappresentato dagli studi di padre Roberto Busa[7], che realizza il primo corpus elettronico delle opere di Tommaso D'Aquino (composto da circa dieci milioni di parole) e un programma per la sua esplorazione, il secondo riguarda invece l'applicazione di metodi formali all'analisi del linguaggio.

Questi metodi si sono sviluppati con l'affermarsi della *grammatica generativa* di Naom Chomsky[8], secondo cui il linguaggio naturale era formato da un sistema di regole strutturate e formulate indipendentemente dall'uso effettivo che si fa della lingua nelle situazioni comunicative. Quindi seguendo il suo approccio, lo sviluppo di un modello computazionale della lingua, per molti anni, ha significato scrivere sistemi di regole linguistiche, per costruire e riconoscere frasi del linguaggio, interpretabili dal computer.

Negli anni '90, però, prende il sopravvento una tradizione di ricerca linguistica basata su una metodologia empirista. Quest'ultima aveva continuato a svilupparsi, parallelamente alla diffusione della grammatica generativa, soprattutto in area anglosassone. L'approccio di Chomsky, basato su sistemi di regole linguistiche, ha portato alla creazione di modelli non capaci di operare in contesti reali. Rispetto alla grammatica generativa con questi metodi di elaborazione di dati empirici, su base probabilistico-statistica, la descrizione del linguaggio era considerata inseparabile dall'analisi dell'uso che si fa di esso.

L'interesse per l'uso di questo approccio ha favorito il diffondersi di corpora come fonte di dati in Linguistica computazionale.

7 Gesuita e linguista italiano e uno tra i pionieri dell'uso dell'informatica applicata alla linguistica
8 Linguista e filosofo americano, fondatore della grammatica generativa

2.1.1 I corpora

I corpora sono raccolte di testi (scritti, orali o multimediali) o parti di essi in formato elettronico, selezionati e organizzati in base a criteri specifici per studiare determinati fenomeni linguistici. Un corpus è un campione rappresentativo di una lingua o di una particolare varietà linguistica (es. linguaggio medico, giuridico, linguaggio infantile ecc).

Il corpus come raccolta di testi, ovviamente in forma cartacea, per lo studio del linguaggio era già una prassi comune prima dell'avvento del computer, quest'ultimo ne ha però favorito la creazione e l'uso, data la sua capacità di raccogliere e immagazzinare una grande quantità di testi sempre crescenti e di ottimizzare la ricerca di dati linguistici.

La costruzione di un corpus non può prescindere da una selezione di testi e ciò va fatto tenendo presente che esso si configura come un campione ragionevolmente accurato della popolazione linguistica in esame, deve quindi essere *rappresentativo* della popolazione stessa, affinché le osservazioni che verranno fatte sul corpus siano generalizzabili all'intera popolazione.

Il progresso tecnologico ha permesso di migliorare i corpora sia dal punto di vista quantitativo (analizzare una grande quantità di testi aumenta le probabilità di osservare i fenomeni che si vogliono studiare) sia da quello qualitativo grazie alle metodologie usate per una migliore selezione dei testi, includendone alcuni ed escludendone altri.

La capacità sempre crescente dei computer di gestire grandi dimensioni di dati ha reso possibile l'uso del Web come fonte di dati per le osservazioni e le analisi linguistiche.

Il Web è la più grande collezione di testi digitali esistente a cui attingere per la costruzione di corpora, i testi disponibili appartengono a un ampia gamma di generi e lingue da utilizzare per creare corpora di vario tipo: generali, specialistici, monolingue, multilingue ecc.

Il rischio di attingere dal web per la creazione di corpora è la diffusione di errori (ortografici, grammaticali o di battitura) che è significativamente più alta rispetto ad altri testi che possono essere raccolti e controllati (E.Corino 2014) , in più bisogna tener conto della quantità di informazione non rilevante che comporta l'uso del web e che deve essere filtrata per avere dati affidabili.

Uno dei vantaggi principali del Web è il continuo aggiornamento che permette di studiare fenomeni linguistici più o meno recenti che non possono essere studiati con i corpora tradizionali.

Per poter sfruttare un corpus come fonte di dati linguistici risulta necessario arricchirlo con informazioni aggiuntive attraverso l'uso di linguaggi di marcatura; questo processo è conosciuto come *annotazione linguistica* e consiste nella codifica di informazione linguistica associata al dato testuale.

Nella linguistica computazionale questa fase di annotazione è fondamentale in quanto permette al computer di interpretare ed esplorare la struttura linguistica implicita del testo .

L'annotazione può essere eseguita manualmente ma più spesso avviene in maniera semi-automatica o automatica, attraverso tecniche di natural language processing, basate su regole o sistemi probabilistici. L'etichettatura linguistica serve per poter elaborare successivamente l'informazione contenuta in un corpus, con lo scopo di ottenere un modello computazionale della lingua che, se applicato a un nuovo insieme di dati, sia in grado di riconoscere il fenomeno linguistico studiato.

2.1.2 Il Natural Language Processing nella LC

Il *natural language processing* (o NLP) è un campo di ricerca riconducibile all'interazione uomo-macchina. La sfida che si pone è quella di riuscire a creare sistemi informatici capaci di interagire con esseri umani tramite l'uso del linguaggio naturale.

La linguistica computazionale e il *natural language processing* sono termini spesso

intercambiabili all'atto pratico anche se il loro scopo sia leggermente differente.

Per aiutare a capire questa differenza basti pensare a WordNet[9] come uno degli obiettivi raggiunti dalla linguistica computazionale e ad Apple Siri[10] come un obiettivo del natural language processing. Entrambi gli obiettivi sono ancora lontani dallo scopo centrale delle due discipline ma la ricerca in questi campi continua ad avanzare.

Esistono diversi strumenti di rappresentazione del testo creati nell'ambito di NLP, utili agli studi di LC e che sono nati ispirandosi a metodologie di LC (un esempio comune è il *POS-tagger*). Questi strumenti sono diventati pratica consolidata nell'analisi automatica del testo e in genere si compongono di quattro fasi che si susseguono a cascata in una pipeline:

- Fase 1: divisione del testo in frasi.
- Fase 2: divisione delle frasi in token (l'unità minima di un testo).
- Fase 3: assegnazione della categoria grammaticale ai singoli token (*POS-tagging*).
- Fase 4: riconduzione di ogni token a lemma, usando la categoria grammaticale nei casi che necessitano di disambiguazione.

Questo è il processo che pone le basi per l'elaborazione del testo che ha bisogno di manipolare gli elementi che lo costituiscono per estrarne aspetti quantitativi (o *features*) utili per successive analisi.

2.2 Machine learning

Al giorno d'oggi il processo di creazione di modelli decisionali per indagini su grandi quantità di dati è affidata prevalentemente a un approccio statistico, fondato su uno dei campi di applicazione dell'intelligenza artificiale: il *machine learning*. Esso

9 http://wordnet.princeton.edu/
10 http://www.apple.com/it/ios/siri/

riunisce metodologie in grado di apprendere in maniera automatica nuova conoscenza a partire da un opportuno insieme di dati iniziali.

Nel 1950 Arthur Samuel, pioniere dell'intelligenza artificiale, diede la prima definizione di *machine learning* (o apprendimento automatico), dicendo che : "l'apprendimento automatico è il campo di studio che dà ai computer l'abilità di apprendere senza essere esplicitamente programmati a farlo".

Più di recente Tom Mitchell, professore presso l'Università di Carnegie Mellon, ha dato una definizione più formale: "Si dice che un programma impara da una certa esperienza E rispetto a una classe di compiti T ottenendo una performance P, se la sua performance nel realizzare i compiti T, misurata dalla performance P, migliora con l'esperienza E" (T. Mitchell 1997).

In altre parole, il *machine learning* si occupa della progettazione e dello sviluppo di sistemi e algoritmi che esaminano iterativamente una grande quantità di dati e riescono a ricavare nuove informazioni direttamente da essi. Tali algoritmi sono capaci di migliorare le loro prestazioni dall'analisi di dati empirici presi in input e generare, tramite l'esperienza maturata, predizioni o classificazioni su nuovi dati.

Esistono diverse tipologie di algoritmi di apprendimento automatico, tra le quali vi sono: s*upervised learning* (o apprendimento supervisionato) e *unsupervised learning* (o apprendimento non supervisionato).

2.2.1 Supervised learning

Il *supervised learning* si pone l'obiettivo di prevedere, dato un elemento di cui si conoscono un insieme di parametri (*features*), il valore di un diverso parametro di output relativo all'elemento stesso. Tale obiettivo viene raggiunto attraverso l'osservazione di esempi con soluzioni (*training set*) che porta alla creazione di un modello che sia capace di predire l'output di nuovi dati.

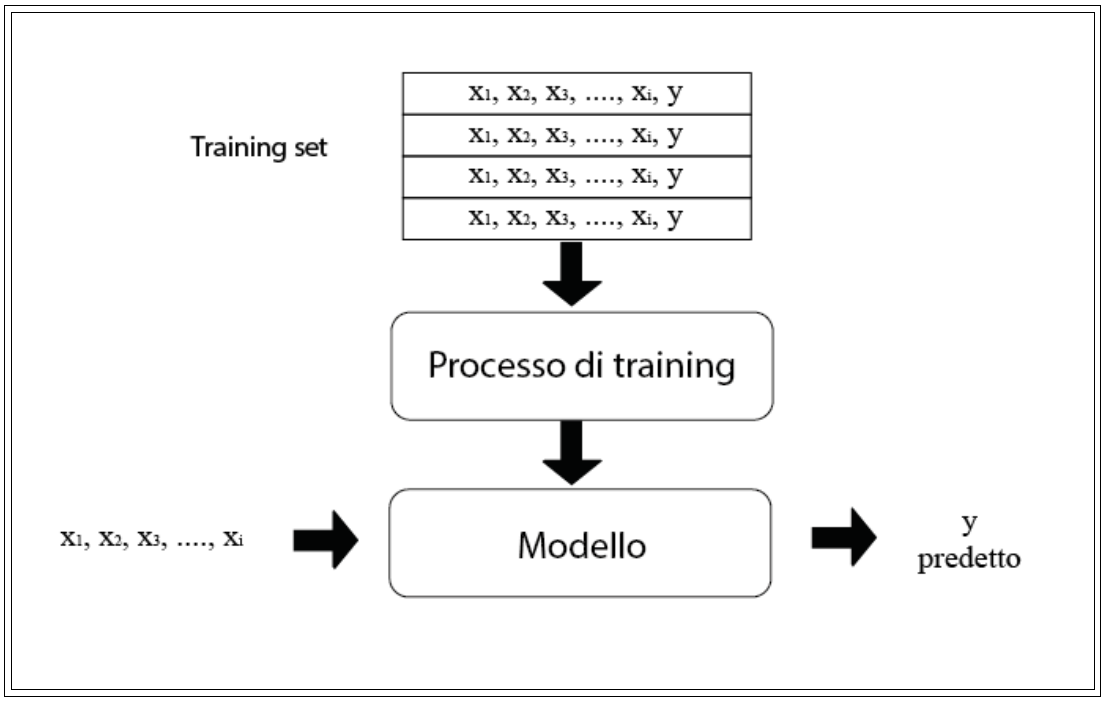


Figura 5. Fasi supervised learning.

Il processo di training cerca di individuare una relazione (sconosciuta) tra l'insieme di *features* di un elemento e l'output ad esso associato, che consenta al modello, dato un nuovo elemento, di predire l'output corretto.

I valori che le *features* possono assumere sono di diverso tipo: quantitativi o qualitativi. Quelli quantitativi specificano la misura di una grandezza, quelli qualitativi specificano una classe di appartenenza.

Anche i valori di output, possono essere di tipo quantitativo o qualitativo. A seconda della tipologia si hanno due generi di predizione differenti. Nel caso in cui l'output è di tipo quantitativo il valore restituito è la predizione di una misura e in questo caso si parla di **regressione**. Se il dato è di tipo qualitativo il valore restituito è l'assegnazione ad una classe e si parla di **classificazione**.

Bisogna sottolineare che la selezione delle *features* spesso aiuta a ridurre l'*overfitting,* un fenomeno che porta il modello ad adattarsi a caratteristiche specifiche solo del *training set,* creando un modello non generalizzabile a un diverso

insieme di dati in input. Una delle cause di questo fenomeno è l'apprendimento fatto attraverso l'uso di troppe *features*, che rischiano di diventare ridondanti portando a un aumento delle prestazioni del modello sui dati di addestramento e, allo stesso tempo, all'aumento dell'errore su un nuovo set di dati.

Il modello ottenuto può essere in seguito valutato, fornendo al sistema un diverso insieme di dati in input (con output noto), e misurando l'accuratezza raggiunta da esso, cioè la percentuale di casi predetti in maniera corretta dal modello.

2.2.2 Unsupervised learning

Unsupervised learning (algoritmi di apprendimento non supervisionato); questa famiglia di algoritmi ha il compito di ricercare strutture e modelli nascosti all'interno dei dati forniti organizzandoli in base ad aspetti e caratteristiche comuni, senza avere un fine esplicito da raggiungere. L'obiettivo è creare un modello che si adatti ai dati, al fine di scoprire proprietà interessanti di essi.

I dati consistono in un insieme di *features* senza un valore di output che funga da riferimento, come invece avviene nel *supervised learning*.

Questo si riflette sulla fase di addestramento dove, essendo fornito al modello un set di dati che non presentano valori che fungono da esempi di soluzione, esso ha bisogno di una quantità di dati in input molto più ampia per riuscire a identificare quelle proprietà che permettono una suddivisione dei dati.

La potenza del *unsupervised learning* è la capacità di individuare connessioni salienti tra i dati che nessun umano penserebbe di cercare, lo svantaggio è che non si hanno indicazioni sul significato di tali connessioni.

2.3 Sentiment Analysis

Con *sentiment analysis* (nota anche come *opinion mining*) si indica l'insieme di tecniche di analisi automatica del testo che hanno lo scopo di identificare ed estrarre opinioni, sentimenti ed emozioni espressi in un testo, rispetto a un argomento, un prodotto o una persona.

L'analisi del sentimento si sviluppa su due obiettivi primari:

- determinare se un testo esprime o meno opinioni (*subjectivity classification*);
- classificare un testo contenente opinioni in positivo o negativo in base all'opinione espressa (*sentiment classification*);

Nel presente lavoro i testi sottoposti ad analisi sono recensioni di applicazioni e si assume che essi esprimano un'opinione. Dunque la *sentiment classification* diventa l'unico aspetto di interesse.

La *sentiment classification* è simile alla classificazione dei testi compiuta nell'ambito dell'information retrieval, che mira a classificare un documento in base all'argomento trattato. Questa classificazione avviene definendo un'insieme di parole (*features*) che rappresentano il topic affrontato nel documento stesso. Nella *sentiment classification* le *features* sono parole usate solitamente per esprimere opinioni, come “bello”, “fantastico”, “orribile”, ecc. L'insieme di *features* viene chiamato lessico o vocabolario.

Il lessico nella *sentiment classification* varia a seconda del dominio in cui le parole sono studiate, non sempre stesse parole esprimono stesse opinioni, ad esempio l'aggettivo “imprevedibile” può assumere un'orientazione negativa nel contesto di recensioni di auto, al contrario, può assumere orientazione positiva se fa rifermento alla trama di un film.

Ad ogni parola viene quindi associato un valore di positività o negatività che può variare in base al contesto. Questa assegnazione produce una scala di valori di sentimento (che vanno dall'estremo negativo all'estremo positivo) che permette di confrontare e aggregare testi anche con stessa orientazione ma con gradazione

differente.

Il Web offre una grande quantità di testi contenenti opinioni (come ad esempio blog, recensioni, commenti, ecc.), questo lo rende infatti il principale campo di applicazione di tali tecniche.

-3-

Metodo e strumenti di ricerca

3.1 Quesiti posti

Il presente lavoro, come già espresso nell'introduzione, mira ad analizzare relazioni tra aspetti di carattere linguistico delle recensioni e la valutazione che gli utenti hanno dato all'applicazione, espressa tramite le apposite stelle.

È stato scelto di studiare l'influenza delle descrizioni sulla valutazione degli utenti e se la scala di valori proposta per la valutazione di un'applicazione soddisfi le necessità degli utenti. Sulla base di ciò sono stati posti i seguenti quesiti:

RQ1) Ci sono aspetti nelle descrizioni di un'applicazione che influenzano l'opinione degli utenti sull'applicazione stessa?

RQ2) Qual è l'uso effettivo che fanno gli utenti della scala di valori quando assegnano una valutazione?

3.2 Metodologia

Per rispondere ai quesiti sopra proposti, è stato deciso di strutturare il metodo di ricerca in due fasi principali: una prima fase di raccolta e preparazione dati e una seconda fase di analisi dei dati e interpretazione dei risultati.

Nella raccolta e preparazione dei dati sono stati creati due corpora, uno composto dalle descrizioni delle applicazioni, e l'altro dalle recensioni ad esse associate. I due corpora sono in formato testuale arricchiti con elementi di marcatura che permettono di ricondurre le recensioni alla relativa descrizione.

Si è proceduto a estrarre automaticamente dal corpus di descrizioni parole chiave (o *keywords*) che rappresentano proprietà distintive di ogni applicazioni, come le sue

funzionalità o elementi che suscitino interesse da parte degli utenti e che, quindi, si suppone siano commentati.

Successivamente il corpus delle recensioni è stato elaborato aggiungendo informazione linguistica di carattere morfologico effettuando *POS-tagging* per ottenere la categoria grammaticale di ogni parola.

In seguito sono state estratte *features* dalle recensioni, che forniscono misure quantitative e qualitative di ogni recensione, come ad esempio: lunghezza, voto (*ratings*), indice di sentimento, presenza di parole chiave, numero di aggettivi, presenza di congiunzioni avversative, ecc. Infine, nella fase di analisi dei dati, sono stati effettuati alcuni esperimenti utilizzando diversi *set di features* e diversi classificatori.

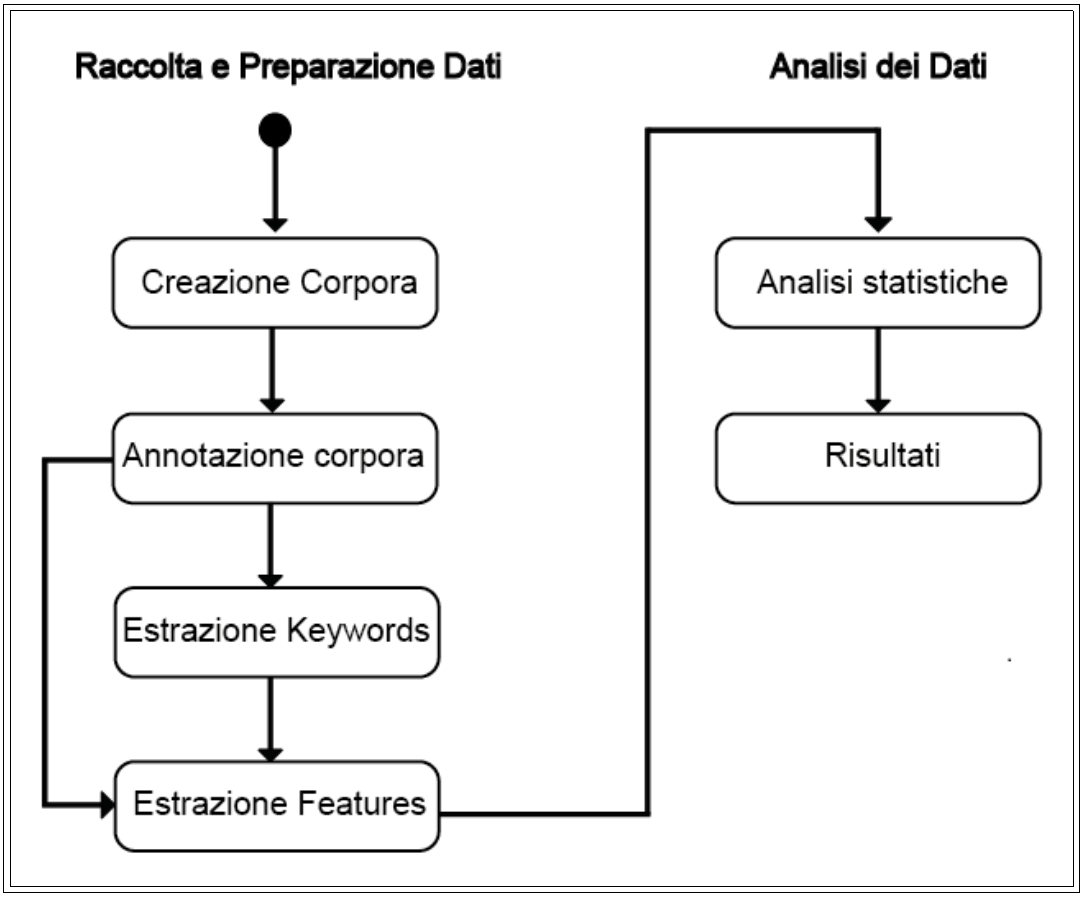


Figura 6. Fasi metodo di ricerca.

Come già detto, questo studio è stato effettuato su recensioni di applicazioni ricavate da Google Play Store, ma con la convinzione che le conclusioni tratte possano essere estese agli altri app store esistenti.

3.3 Strumenti utilizzati

In questo capitolo sono presentati i principali strumenti che sono stati utilizzati nel corso del progetto, necessari per l'indagine svolta. Tali strumenti sono:

- Python: linguaggio di programmazione;
- un dizionario generico di parole polarizzate;
- Tanl: suite di strumenti per l'analisi del testo;
- Weka: software per l'apprendimento automatico.

3.3.1 Python

Python è un linguaggio di programmazione ideato da Guido Van Rossum, un informatico olandese, all'inizio degli anni novanta e rilasciato per la prima volta nel 1991.

Si tratta di un linguaggio di programmazione pseudocompilato ad alto livello, nato con lo scopo di essere semplice, intuitivo e facilmente comprensibile, infatti il codice risulta molto simile all'inglese parlato. Supporta diversi paradigmi di programmazione come quello imperativo, funzionale e orientato agli oggetti, si può quindi definire un linguaggio multiparadigma.

È un linguaggio a tipizzazione dinamica, i tipi delle variabili non devono essere dichiarati in anticipo, come avviene in altri linguaggi di programmazione (ad esempio C o JAVA), dato che il controllo del tipo è effettuato a *runtime* e non in fase di compilazione.

La sintassi risulta molto semplice, per i blocchi logici non si usano parentesi o parole chiave ma vengono costruiti sfruttando l'indentazione, questo migliora la leggibilità ed è utile soprattutto quando diversi autori lavorano sullo stesso codice.

Un altro vantaggio di questo linguaggio è la vasta libreria standard che lo rende adatto a molti impieghi, anche per questo è uno dei linguaggi più ricchi e comodi da usare.

3.3.2 Dizionario generico di parole polarizzate

Questa risorsa comprende circa 7100 lemmi idi parole italiane polarizzati. Tale risorsa è stata derivata da SentiWordNet[11] in inglese di A. Esuli e F. Sebastiani.

Il dizionario presenta su ogni riga un lemma con associato il suo valore di positività e il suo valore di negatività. Questi valori sono numeri compresi tra 0 e 1, dove 0 indica l'assenza di componente positiva o negativa, mentre 1 rappresenta il massimo valore di positività o negatività.

Come detto in sezione 2, in questi tipi di risorse i valori attribuiti ai lemmi dipendono dal contesto in cui vengono prodotti, questa risorsa specifica è stata prodotta con lo scopo di rappresentare un contesto generale.

3.3.3 Tanl Pipeline

Tanl[12] (text analytics and natural language) è una suite di strumenti per l'analisi del testo, sviluppata da MediaLab[13] presso il dipartimento di Informatica dell'Università di Pisa.

La suite è disponibile attraverso un servizio web che permette di estrarre *Named Entities* da un testo e di produrre gli alberi sintattici delle frasi presenti in esso.

11 http://sentiwordnet.isti.cnr.it/
12 http://tanl.di.unipi.it/it/pipe.html
13 http://medialab.di.unipi.it/

Questo servizio utilizza cinque moduli dalla suite che servono per:

- dividere un testo in frasi;
- tokenizzare le frasi;
- estrarre il lemma, il POS e la morfologia per ogni token;
- estrarre Named Entities;
- costruire l'albero sintattico di dipendenze.

I moduli sono presentati attraverso una pipeline che può essere richiamata anche programmaticamente.

Il servizio è disponibile per testi in italiano, in inglese e in spagnolo.

3.3.4 Weka

Weka[14] (waikato environment for knowledge analysis) è un software open source che è stato ideato e sviluppato al Dipartimento di Computer Science dell'Università di Waikato in Nuova Zelanda. Si tratta di un software interamente scritto in Java per l'apprendimento automatico.

Weka metta a disposizione una vasta collezione di algoritmi per applicazioni di Data Mining.

La sua interfaccia grafica è composta da:

- Explorer, un ambiente per l'esplorazione dei dati.
- Experiment, il quale permette di eseguire degli esperimenti e condurre test statistici tra sistemi di apprendimento.
- KnowledgeFlow, una variante dell'ambiente Explorer; qui le operazioni da eseguire si esprimono, però, in un ambiente grafico e permette in più una gestione dei dati in modo incrementale.

14 http://www.cs.waikato.ac.nz/ml/weka/

- Simple CLI, il quale consente l'accesso a tutte le classi di Weka usando il software da riga di comando.

Dall'ambiente Explorer (vedi figura 7) si possono adoperare diversi tipi di lavori, corrispondenti ai bottoni presenti nell'interfaccia:

- Preprocess: serve per importare e preparare i dati.
- Classify: permette di applicare ai dati gli algoritmi per la classificazione e i modelli per la regressione.
- Cluster: serve per fare analisi di cluster.
- Associate: serve per applicare gli algoritmi di apprendimento delle regole di associazione.
- Select Attributes: permette di selezionare sottogruppi di attributi per l'analisi.
- Visualize: utile per visualizzare le proprietà grafiche dei dati.

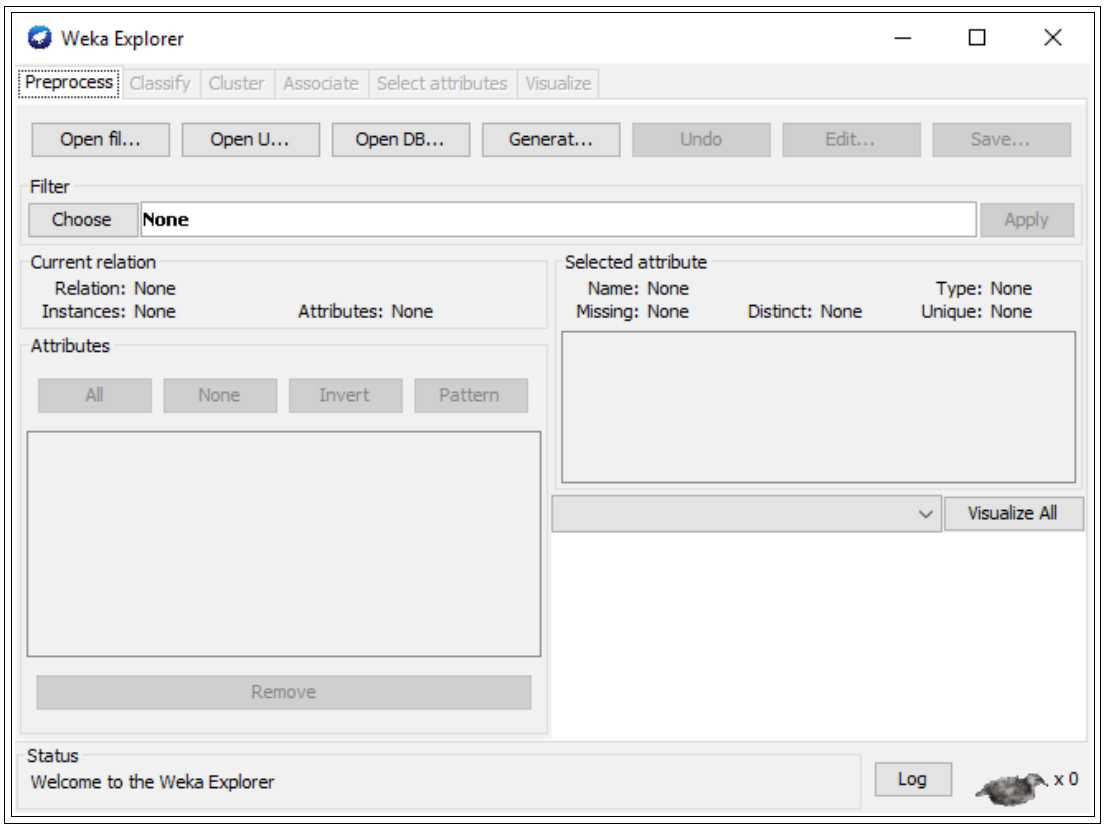


Figura 7. Interfaccia Weka.

Weka possiede le implementazioni delle principali tecniche per la classificazione e per la regressione come alberi di decisione, classificatori Bayesiani, support vector machines, logistic and linear regression ecc.

Nella sezione Classify si possono scegliere il tipo di classificatore da utilizzare e il tipo di metodo con cui vogliamo valutare la performance del classificatore selezionato e vedere i dettagli riguardanti la sua performance nel riquadro grande a destra (vedi Figura 8).

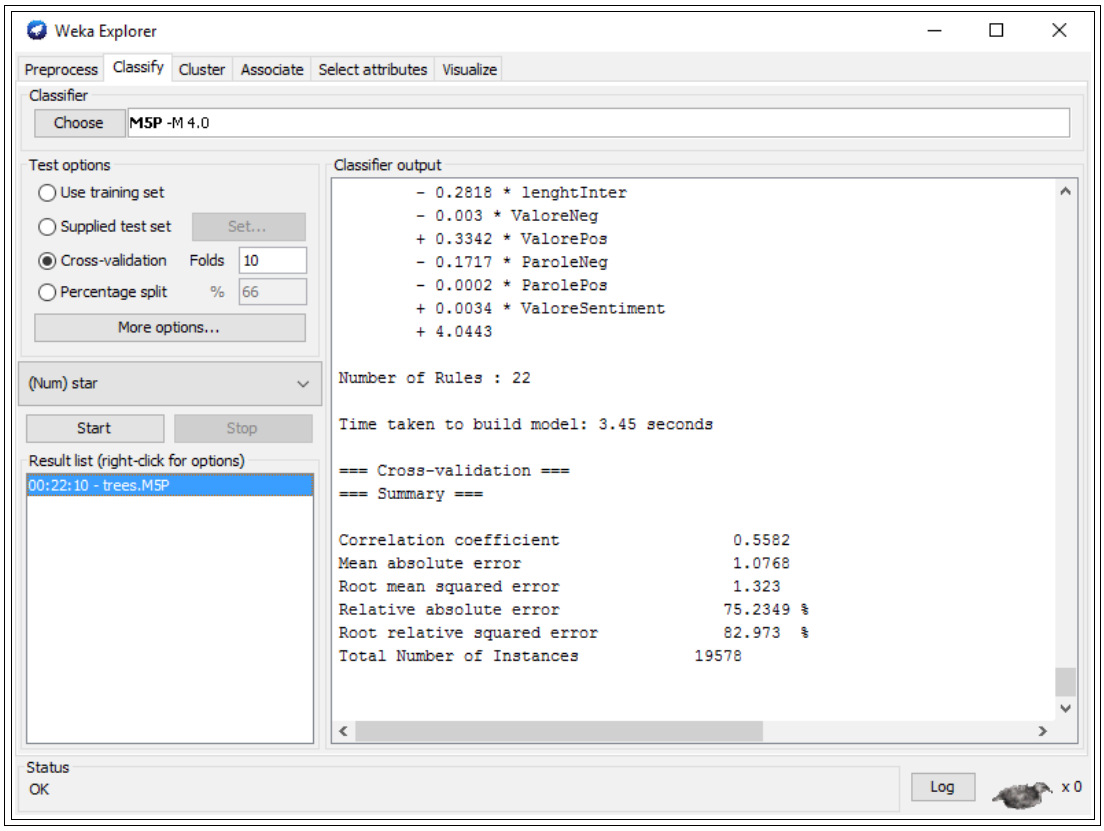


Figura 8. Sezione Classify Weka.

3.3.5 File ARFF

Per caricare il set di dati in Weka il formato utilizzato è ARFF (Attribute-Relation File Format), sviluppato sempre nell'Università di Waikato con lo scopo di essere usato come input per l'analisi dei dati con Weka.

Si tratta di un formato testuale utilizzato per memorizzare i dati in un database. Tali file descrivono la relazione, gli attributi e i valori che questi possono contenere.

I tipi di dati supportati sono:

- Numeric (i numeri possono essere sia reali sia interi).
- String.
- Nominal.
- Date (rappresenta il formato data).

```
@RELATION iris

@ATTRIBUTE sepallength    NUMERIC
@ATTRIBUTE sepalwidth    NUMERIC
@ATTRIBUTE petallength    NUMERIC
@ATTRIBUTE petalwidth    NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-versicolor
5.0,3.6,1.4,0.2,Iris-versicolor
5.4,3.9,1.7,0.4,Iris-versicolor
4.6,3.4,1.4,0.3,Iris-virginica
5.0,3.4,1.5,0.2,Iris-virginica
4.4,2.9,1.4,0.2,Iris-virginica
4.9,3.1,1.5,0.1,Iris-virginica
```

Figura 9. Esempio file con formato .arff.

Un file Arff, come si può vedere in figura 9, è composto da due sezioni distinte: *Header* e *Data*. L'*Header* contiene il nome della tabella, una lista di attributi, e il tipo di valori che possono assumere.

La sezione *Data* contiene la riga di dichiarazione dei dati e le righe con le effettive

istanze. Ogni istanza è rappresentata su una singola linea e i valori, separati da una virgola, devono apparire nell'ordine in cui sono stati dichiarati nella sezione *Header,* ogni valore corrisponde all'attributo che si trova nella stessa posizione dell'intestazione.

3.3.6 M5P Regression Tree

L'obiettivo della regressione, come detto in sezione 2, è predire un valore quantitativo definendo la relazione che intercorre tra i valori forniti in input: *features* e relativi output.

I *regression tree*, come i *decision tree*, utilizzano una rappresentazione ad albero dei dati. Le *features* sono poste sui nodi dell'albero, mentre sugli archi sono definite regole di condizione (if, then) che, a seconda del valore che assume una *feature*, può portare a differenti nodi o direttamente alle foglie dell'albero dove è espressa la predizione (vedi figura 10).

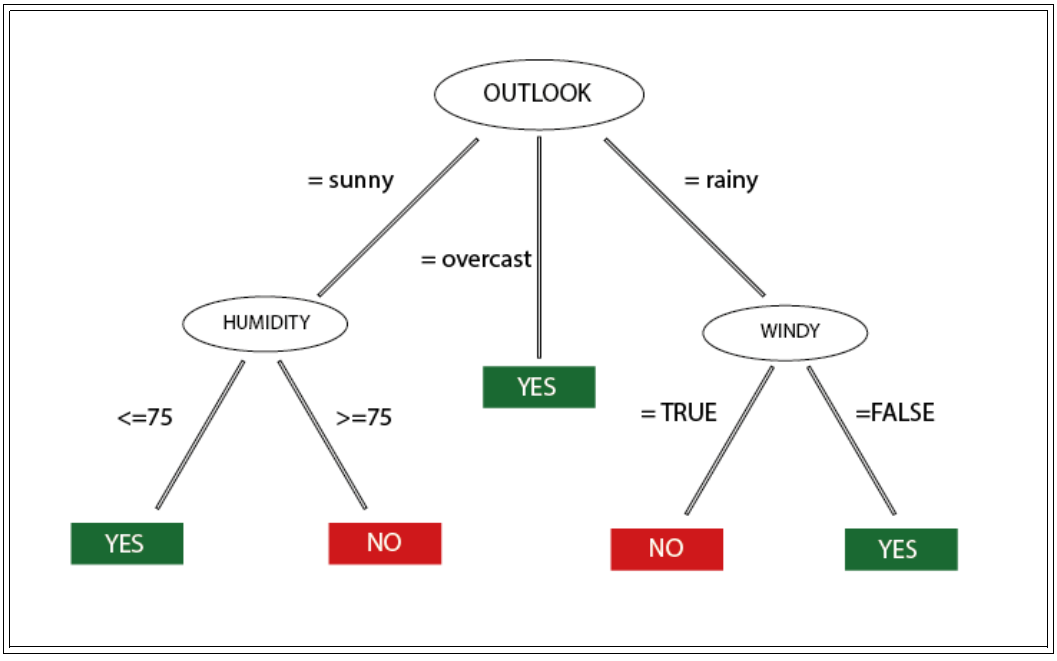


Figura 10. Esempio di decision tree (Quinlan J.R., *Induction of Decision Trees, 1986*).

La differenza sostanziale tra *regression tree* e *decision tree* risiede nello stabilire il valore di una foglia: nel caso del decision tree si sceglie la classe maggioritaria, nel caso del *regression tree* tale valore è la funzione di regressione lineare dei valori delle istanze di input considerate per il ramo dell'albero da cui deriva la foglia.

Il *regression tree* più utilizzato è M5P, ricostruzione dell'algoritmo M5 di Quinlan, il quale è stato impiegato in questo lavoro, in quanto gli alberi di regressione hanno il vantaggio, a differenza di alti generi di classificatori, di fornire insieme al modello, anche regole facilmente interpretabili che hanno portato alla creazione di esso. Tale scelta risulta appropriata al voler conoscere le relazioni che intercorrono tra le *features* (input) e il numero di stelle (output), scopo di questo lavoro.

L'apprendimento attraverso l'algoritmo M5P prevede due fasi: una prima fase dove viene ottenuto l'albero che meglio descrive i dati a partire dalla costruzione dei possibili alberi ricavabili, e scegliendo tra essi il miglior compromesso tra minor dimensioni dell'albero e capacità descrittiva; una seconda fase di “potatura” per ridurre le dimensioni dello stesso, ottenendo un minor costo computazionale e una migliore interpretabilità delle regole, a discapito dell'accuratezza anche se in maniera poco significativa.

-4-

Fase 1: raccolta e preparazione dati

4.1 Creazione dei corpora

Inizialmente si è cercato di reperire un corpus di recensioni di applicazioni in italiano, ma non essendo disponibile un corpus di questo tipo è stato necessario crearlo. Tra i vari *app store* presenti nel mercato (Google Play Store, Windows Phone Store e Apple App Store) si è deciso di raccogliere i dati esclusivamente dallo store di Google. Di conseguenza, anche il corpus delle descrizioni è composto interamente da descrizioni di applicazioni reperite da questo store. Attualmente non esiste un API per accedere al contenuto di Play Store con facilità, si è quindi creato un programma ad hoc che utilizza tecniche di *Web Scraping* per raccogliere dati dalla pagina dello store che sono stati salvati in file di testo.

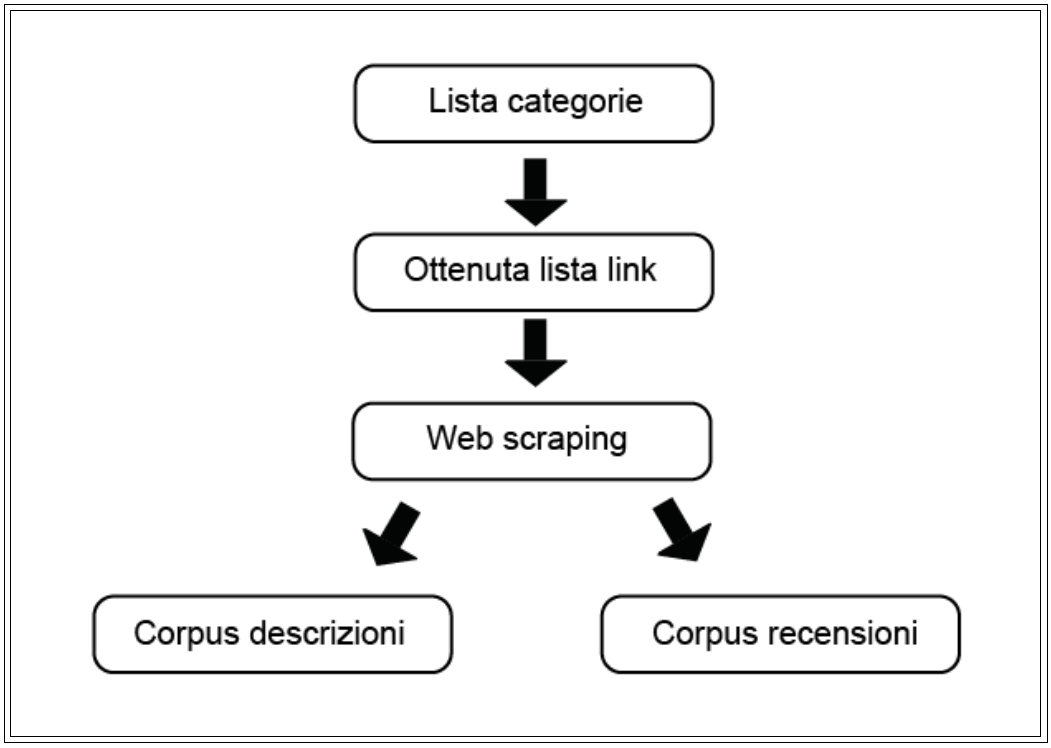


Figura 11. Fasi costruzione corpora.

Questo programma, interamente scritto in Python, prende in input una lista di link ad applicazioni dello store. Per ottenere questa lista sono state scelte manualmente 7 categorie (affari, comunicazione, finanza, notizie e riviste, viaggi e info locali, shopping, libri e consultazione) tra quelle proposte da Play Store; per ogni categoria sono state prese automaticamente le 30 applicazioni più popolari, 15 a pagamento e 15 gratuite, per ottenere alla fine del processo, un corpus bilanciato e rappresentativo dell'uso della lingua nell'ambito delle recensioni di applicazioni.

Il primo passo è stato, dunque, ottenere questa lista con 210 url di applicazioni attraverso l'ausilio di uno script. La lista è stata, successivamente, utilizzata dal programma in maniera iterativa per effettuare una richiesta *get* HTTP per ogni url. In questo modo per ogni applicazione è stata ottenuta in risposta la pagina HTML contenente la descrizione e 40 recensioni.

Per avere un corpus di dimensioni soddisfacenti era stato deciso di recuperare 200 recensioni per applicazione. Le restanti recensioni, dunque, sono state raccolte con l'utilizzo di richieste *post* HTTP, imitando il comportamento di un utente che scorre le recensioni di un'applicazione usando l'apposito bottone presente nella pagina.

Le risposte alle richieste HTTP contengono l'intero documento HTML della pagina. I documenti HTML sono file di testo etichettato con tag di marcatura che indicano, tra le altre cose, la struttura gerarchica del testo. Tale struttura è riconducibile a quella di un albero dove ogni tag rappresenta un nodo. Esistono diverse librerie per Python che permettono di navigare un documento HTML sfruttando la sua struttura ad albero, quella che è stata usata è *lxml*[15]. Essa è pensata prevalentemente per file XML ma adattabile con facilità a lavorare su file HTML data la similarità.

Con l'ausilio di questa libreria è stato possibile ricostruire la struttura ad albero dalle stringhe di testo in risposta alle richieste HTTP, e andare a recuperare il contenuto dei nodi di nostro interesse tramite il metodo *xpath,* messo a disposizione dalla libreria stessa.

Una volta raggiunti i nodi si è resa necessaria una pulizia dei dati. Sono stati rimossi gli elementi HTML tramite l'uso di espressioni regolari e sono state convertite le

15 http://lxml.de/

entità HTML (o *HTML entities*). Gli *HTML entities* sono entità che sono pensate per permettere di inserire in un testo caratteri riservati per la sintassi HTML (es. “<”) ed anche caratteri particolari come alcune lettere accentate (ad esempio “ý”) . Quest'ultimo aspetto però è stato in parte risolto con l'adozione di UTF-8 come codifica di default per i documenti HTML.

Ogni entità HTML è stata sostituita con il carattere corrispondente utilizzando il modulo standard di Python *HTMLParser*.

Dopo questa breve fase le informazioni relative alle descrizioni e quelle relative alle recensioni sono state poi salvate in due file differenti impostando il sistema di codifica utf-8.

4.1.1 Corpus descrizioni

Per la creazione del corpus di descrizioni sono stati selezionati i nodi della pagina HTML (vedi figura 12) relativa alla descrizione di ogni applicazione contenenti le informazioni ritenute necessarie per l'indagine: il testo della descrizione e il voto medio dell'applicazione (vedi figura 13). Quest'ultimo è la media dei voti dati dagli utenti nelle recensioni e quindi un'applicazione con assenza di valutazioni non ha un voto medio associato. In questo caso non viene generato l'elemento HTML che racchiude il voto dell'applicazione.

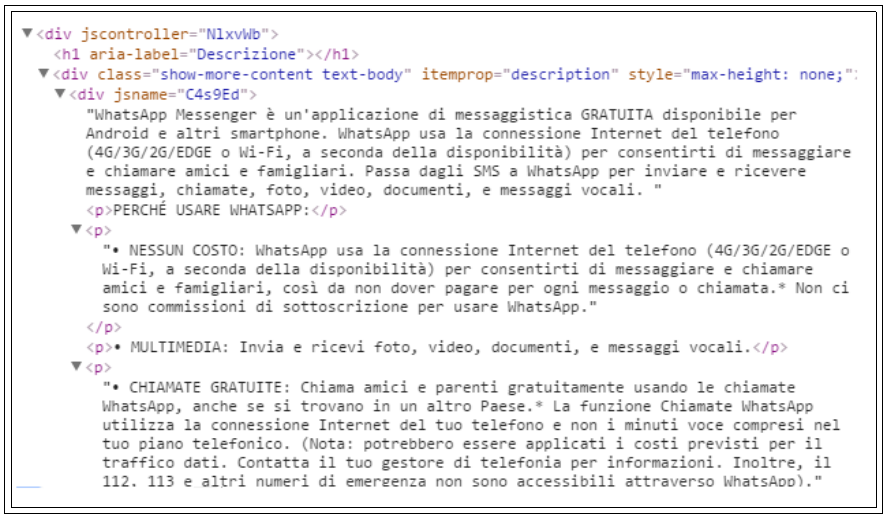


Figura 12. Frammento descrizione pagina HTML .

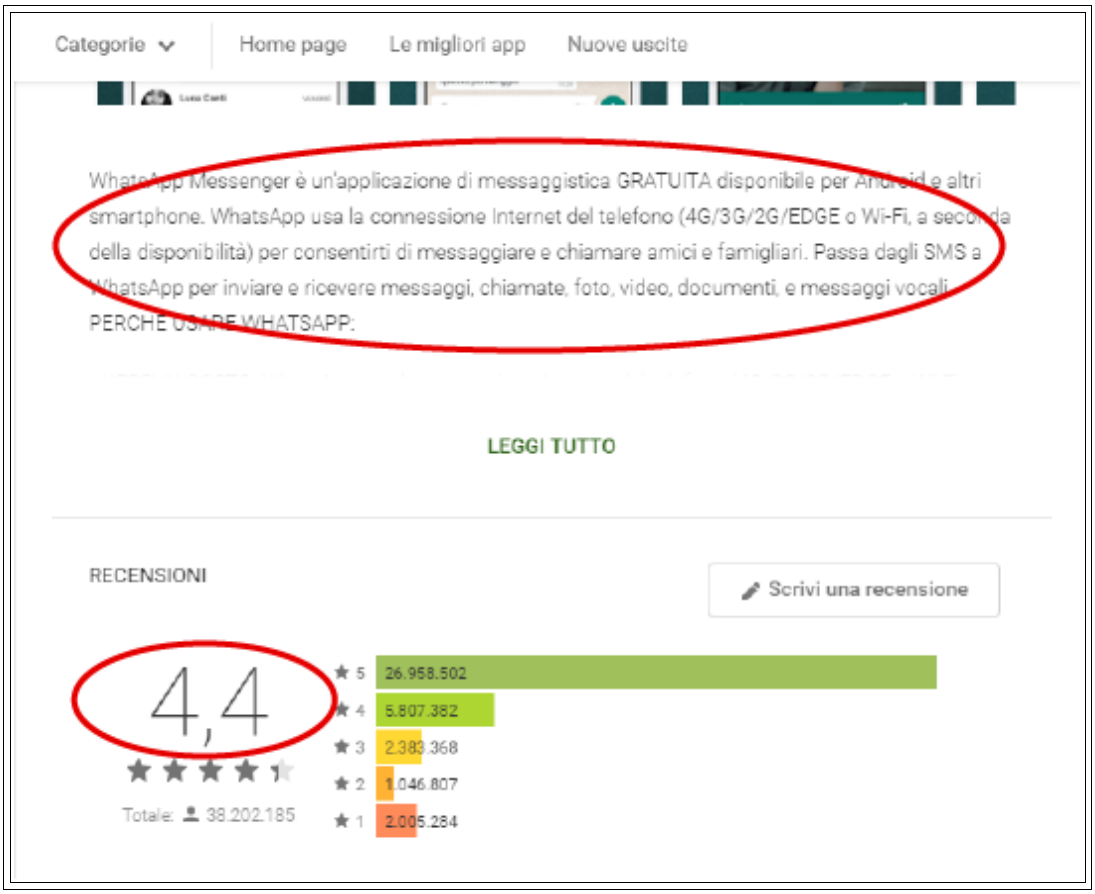


Figura 13. Frammento della pagina contenente una descrizione da Google Play Store .

Il corpus, come si può vedere in figura 14, è un file di testo al cui interno le informazioni sono state suddivise da etichette, similmente ai file XML. La prima tag `<ROOT>` identifica il nodo radice e racchiude l'intero insieme di descrizioni. Ogni descrizione è racchiusa tra le tag `<description></description>` e a sua volta contiene all'interno le tag `<id></id>` in cui è riportato l'id assegnato nel Play Store che compone parte dell'URL dell'applicazione. Sempre all'interno delle tag `<description></description>` sono presenti le tag `<Star></Star>` in cui è riportato il voto medio dell'applicazione e le tag `<text></text>` in cui è racchiuso il testo della descrizione.

Sebbene Google Play Store sia localizzato anche in italiano, è possibile incontrare descrizioni di applicazioni in lingue diverse dall'italiano, tipicamente inglese, che è la

lingua predefinita per Play Store. Questo dipende dal fatto che la localizzazione di descrizione e altre informazioni relative a un'applicazione sono affidate a chi produce l'applicazione. L'unico servizio che Google Play mette a disposizione agli sviluppatori è un software di traduzione automatica. Di conseguenza, nel corpus si possono trovare sia descrizioni in lingua diversa dall'italiano seguite dalla traduzione in italiano generata automaticamente, sia descrizioni esclusivamente in lingua straniera, e per la maggior parte descrizioni in lingua italiana. All'interno del corpus le descrizioni con doppia lingua sono state divise attraverso l'uso della tag `<CL>` che indica il passaggio da una lingua all'italiano all'interno del nodo `<description>`.

```
<root>
<description>
<id>eu.amway.mobile.businessapp</id>
<star>4,1</star>
<text>L'applicazione è pensata esclusivamente per gli Imprenditori Amway attivi i
</description>
<description>
<id>com.piksoft.turboscan</id>
<star>4,7</star>
<text> TurboScan turns your phone into a full-featured and powerful multipage sca
</description>
<description>
<id>com.ospolice.packagedisablerpro</id>
<star>4,3</star>
<text>***This application works only on Samsung devices *** NO ROOTING REQUIRED
</description>
<description>
<id>com.dynamixsoftware.printershare.premium</id>
<star>4,5</star>
<text>Questa chiave sblocca le funzionalità premium dell'app gratuita PrinterShar
</description>
<description>
<id>co.hotplate.instalogo</id>
<star>4,2</star>
<text>Localized in 34 languages! Use this app in your native language!  Supported
</description>
```

Figura 14. Frammento corpus descrizioni.

4.1.2 Corpus recensioni

La creazione del corpus delle recensioni segue la stessa procedura del corpus sopra descritto, le informazioni raccolte per ogni recensione si limitano al voto assegnato dall'utente e al testo della recensione.

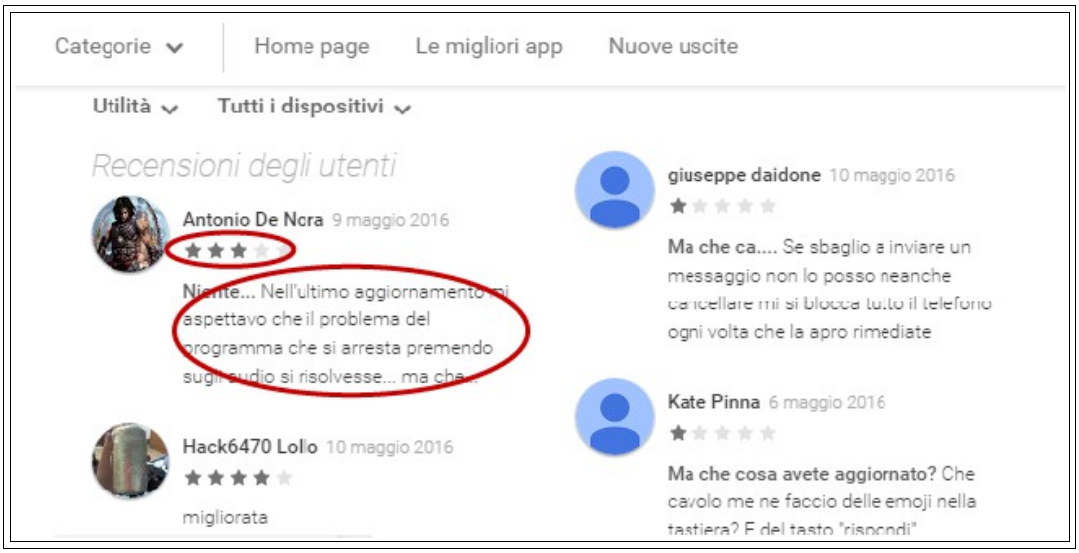


Figura 15. Frammento della pagina contenente recensioni da Google Play Store .

Anche questo corpus si presenta come un file di testo (vedi figura 16) suddiviso da etichette di marcatura. Oltre al nodo radice `<ROOT>`, troviamo il nodo `<review>` al cui interno è presente sia il testo della recensione, tra le tag `<text></text>`, sia il numero di stelle racchiuso tra le tag `<Star></Star>`.

Ogni nodo `<review>` inoltre presenta al suo interno l'id dell'applicazione a cui fa riferimento tra le tag `<id></id>`. Tramite questa etichetta è quindi possibile ricondurre ogni recensione alla descrizione dell'applicazione che recensisce.

```
<id>livio.pack.lang.en_US&hl=it</id>
<star>5</star>
<text>Una delle migliori!!  Eccellente app. Fornisce diversi significati per ogni parola, presenti
</review>
<review>
<id>livio.pack.lang.en_US&hl=it</id>
<star>4</star>
<text>Dizionari precisi...  Tutti i dizionari offline sono precisi, Livio, ma vorrei che mettessi
</review>
<review>
<id>livio.pack.lang.en_US&hl=it</id>
<star>2</star>
<text>Mancano parole, scaricate DIC dizionario  Molte parole mancano e durante lo studio dell'ingl
</review>
<review>
<id>livio.pack.lang.en_US&hl=it</id>
<star>5</star>
<text>App ottima  Potresti aggiungere come app anche il dizionario di Latino? Sarebbe un successon
</review>
<review>
<id>livio.pack.lang.en_US&hl=it</id>
<star>5</star>
<text>Everyday  La uso tutti i giorni per semplicità, ricchezza e comodità offline.</text>
</review>
```

Figura 16. Frammento corpus recensioni.

Le recensioni sono quasi tutte in italiano, eccetto alcuni casi in cui gli utenti commentano in una lingua differente da quella di localizzazione, generalmente in lingua inglese. Quindi è possibile incontrare, anche se in numero molto ristretto, recensioni in lingua diversa dall'italiano.

Il numero di recensioni nel corpus stimato era di 42000 (210 applicazioni * 200 recensioni), ma alla fine del processo di creazione del corpus questo numero si è ridotto notevolmente arrivando a 23876.

Una prima causa è addebitabile al numero di recensioni di alcune applicazioni che non sempre ha raggiunto 200, tipicamente si trattava di applicazioni a pagamento. Un'altra motivazione è il filtro inserito nel programma che scarta tutte le recensioni con un numero di caratteri minore a 20. Questo filtro è stato inserito in quanto si è ritenuto che probabilmente una recensione così breve non sarebbe stata informativa e dunque non utile per la nostra analisi.

Dopo questa etichettatura generale, il corpus delle recensioni è stato annotato (vedi figura 17), utilizzando il servizio online di Tanl (vedi sezione 3), aggiungendo informazioni di tipo morfologico e sintattico. Al termine di questo processo, il testo

delle recensioni contenuto nel corpus è stato profondamente modificato nell'aspetto: ogni riga contenuta all'interno della tag `<text></text>` corrisponde ad un singolo token della frase con gli attributi che lo descrivono per ogni colonna. Questa rappresentazione testuale è definita dal formato CoNLL[16].

Gli attributi sono:

- la prima colonna è l'id del token, parte da 1 per ogni nuova frase;
- la seconda colonna indica la forma del token, cioè come è stato riscontrato nel testo grezzo;
- la terza colonna indica il lemma a cui risale il token;
- la quarte colonna rappresenta la categoria grammaticale a cui appartiene il token (ad esempio “R” indica l'articolo);
- la quinta colonna rappresenta la categoria grammaticale a un livello più profondo (ad esempio “RD” indica l'articolo determinativo);
- la sesta colonna rappresenta l'informazione morfologica, quando specificata, di genere e numero o modo e tempo se riferita a verbi;
- la settima colonna indica l'id di un altro token con cui il token della riga ha una relazione di dipendenza;
- l'ottava colonna indica il tipo di relazione di dipendenza o “ROOT” nel caso in cui il token sia la radice della frase.
- le ultime due colonne sono lasciate vuote.

16 http://ifarm.nl/signll/conll/

```
<root>
<review>
<id>eu.amway.mobile.businessapp&hl=it</id>
<star>4</star>
<text>
1   Errore  errore  S   S   num=s|gen=m 3   subj_pass   _   _
2   Ha  Ha  V   VA  num=s|per=3|mod=i|ten=p 3   aux _   _
3   funzionato  funzionare  V   V   num=s|mod=p|gen=m   0   ROOT    _   _
4   correttamente   corretto    B   B   _   3   mod _   _
5   per per E   E   _   3   comp    _   _
6   tre tre N   N   _   7   mod _   _
7   giorni  giorno  S   S   num=p|gen=m 5   prep    _   _
8   ,   ,   F   FF  _   3   con _   _
9   poi poi B   B   _   11  mod _   _
10  mi  mi  P   PC  num=s|per=1|gen=n   11  comp    _   _
11  evidenzia   evidenziare V   V   num=s|per=3|mod=i|ten=p 3   conj    _   _
12  le  il  R   RD  num=p|gen=f 15  det _   _
13  problematiche   problematico    A   A   num=p|gen=f 15  mod _   _
14  già già B   B   _   15  mod _   _
15  riscontrate riscontrare V   V   num=p|mod=p|gen=f   11  arg _   _
16  dagli   da  E   EA  num=p|gen=m 15  comp    _   _
17  altri   altro   P   PI  num=p|gen=m 16  prep    _   _
18  ,   ,   F   FF  _   15  punc    _   _
19  come    come    C   CS  _   27  mod _   _
20  posso   potere  V   VM  num=s|per=1|mod=i|ten=p 21  aux
```

Figura 17. Frammento corpus recensioni annotato.

4.1.3 Dati corpora

I due corpora presentano ovviamente dimensioni molto diverse. Il corpus delle descrizioni è composto da 87.438 token; i lemmi unici sono 8.891 (dimensione del vocabolario).

Il corpus delle recensioni, ben più grande, si compone di 546.353 token, di cui 31.317 parole tipo (o *type*) e 23.540 lemmi unici. Questo corpus nella fase di analisi è stato sottoposto ad una analisi testuale più profonda ed è bene quindi riportare alcune informazioni quantitative che ne rappresentano la ricchezza lessicale (vedi tabella 1). Per fare ciò è stato calcolato il *type token ratio* (o TTR) e la percentuale di token con frequenza pari a 1 (detti *hapax*).

Numero token	546.353
Numero type	31.317
Vocabolario	23.540
Percentuale *hapax*	8%
TTR	0,05

Tabella 1. Dati linguistici corpus recensioni.

Il TTR è il rapporto tra il numero di parole tipo presenti in un testo e il numero di token che lo compongono. È un valore compreso tra 0 e 1 e indica quanto vario è il testo rispetto alle parole utilizzate. Più il valore si avvicina a 1 più il testo presenta varietà nelle forme.

Un altro modo per esprimere la varietà delle forme di un testo è la percentuale di *hapax*, cioè forme con frequenza unica.

Come si può vedere dalla tabella numero 1 la varietà lessicale, cioè la ricchezza, è estremamente bassa in questo corpus. Nonostante le applicazioni spazino nei temi e negli argomenti, gli utenti quando le commentano utilizzano un vocabolario molto ristretto. Ciò è dovuto sicuramente al fatto che l'italiano utilizzato è riconducibile ad un solo dominio specifico (recensioni di applicazioni).

4.2 Identificazione delle keywords

Il corpus delle descrizioni, come già detto, funge da fonte di dati da cui reperire informazioni rappresentative di un'applicazione ipotizzando che questi aspetti siano presenti nei commenti degli utenti.

Il metodo qui descritto propone di identificare un insieme di parole all'interno del

testo della descrizione che esprimono proprietà, funzionalità o aspetti caratterizzanti di un'applicazione.

Ad esempio per l'applicazione di *LinkedIn Job Search* alcune delle parole che sono state recuperate sono: “lavoro”, “ricerca”, “opportunità”. Essendo un'applicazione utile per farsi conoscere nel mondo del lavoro, già queste tre parole riflettono qual è la sua funzionalità. A differenza di applicazioni simili, che però fungono esclusivamente da portale per gli annunci di lavoro, LinkedIn permette di avere un proprio spazio dove rendere visibili le proprie informazioni e competenze; infatti è l'unica applicazione di questo genere che presenta anche la parola “profilo”, la quale indica un suo aspetto caratterizzante.

Per recuperare questo insieme di parole da ogni descrizione si è deciso di agire calcolando per ogni parola il TF-IDF.

Data una collezione di testi, il TF-IDF è un valore che indica la rilevanza di una parola in un testo rispetto al numero di testi della collezione in cui non compare. Più alto è il valore più la parola è considerata rilevante.

Questa valore è la combinazione di due misure: il TF (*term frequency*) e l'IDF (*inverse document frequency*).

$$TF(i,j) = \frac{freq(i,j)}{maxfreq(i,j)}$$

$$IDF(i) = \log \frac{N}{n(i)}$$

$$TF\text{-}IDF(i,j) = TF(i,j) * IDF(i)$$

Figura 18. Formula TF-IDF .

Il *term frequency* misura la frequenza di una parola in un testo rispetto alla frequenza della parola più frequente nel testo. Lo scopo è quello di dare un valore di frequenza ad una parola che sia comparabile tra testi di lunghezza differente. La frequenza di un termine è spesso usata come indicatore di importanza della parola stessa.

L'*inverse document frequency* è dato dal logaritmo del rapporto tra il numero di testi presenti nella collezione e il numero di testi in cui la parola compare. Questo valore serve ad indicare quanto una parola è caratterizzante per un testo: una parola che compare in tutti i testi della collezione difficilmente sarà rappresentativa di uno soltanto di questi.

Entrambi i valori mirano a stabilire quanto una parola è rilevante, il primo dalla prospettiva di un unico testo, il secondo dalla prospettiva dell'intera collezione di testi. La combinazione di queste due misure tiene conto di entrambi gli aspetti.

La misura del TF-IDF è una tecnica nata nell'ambito dell'Information Retrival e serve a fornire una facile rappresentazione di un documento testuale. Essa risulta adatta all'indagine proposta nel presente lavoro, assumendo che le descrizioni siano informative per la loro applicazione.

4.2.1 Processo di identificazione delle keywords

Prima di poter calcolare il TF-IDF delle parole che compongono il corpus delle descrizioni è stata effettuata una fase di *pre-processing* dei dati. Come già detto, nel corpus sono presenti descrizioni in lingue diverse. Per prima cosa, quindi, si è reso necessario un processo di riconoscimento della lingua. Questo è stato realizzato mediante un confronto tra le parole presenti nel testo e un set di stopwords di diverse lingue, questi set sono messi a disposizione dalla libreria *nltk*[17]. Sono state scartare tutte le descrizioni in lingua diversa dall'italiano e dall'inglese.

In seguito sono state riportate al loro lemma tutte le parole che compongono la descrizione utilizzando la suite di Tanl (vedi sezione 3). A questo punto è stato

17 http://www.nltk.org/

possibile rimuovere stopwords, punteggiatura e simboli.

Una volta conclusa la fase di *pre-processing* si è calcolato il TF-IDF dei lemmi rimasti, ottenendo così una lista di lemmi associati al loro valore. Da questa lista è stata selezionata una parte contenente i lemmi con valore più alto. Il numero di lemmi appartenenti a questo sottoinsieme è stato scelto sulla base di verifiche manuali.

Poiché una descrizione è un testo relativamente breve (Play Store impone una lunghezza massima di 4000 caratteri) otto lemmi sono stati considerati sufficienti per rappresentare l'applicazione e ridurre al minimo la presenza di parole non rappresentative.

Con lo stesso intento di ottenere un insieme di parole chiave più accurato si è deciso di rimuovere dalla lista dei lemmi dell'intero corpus tutti quei lemmi con una frequenza minore di 3 in tutto il corpus. Questo perché è stato notato che nell'insieme risultante di parole chiave, apparivano spesso token che erano url a pagine web. A questi token il TF-IDF aveva assegnato un valore alto ma ai fini della nostra analisi non erano utili. Applicando questo filtro tutti quei token che avevano un'unica occorrenza in tutto il corpus, o al massimo due nei casi delle descrizioni con doppia lingua, sono stati selezionati e rimossi. Questo ha permesso un miglioramento nel risultato.

4.3 Estrazione delle features

L'ultimo passaggio nella fase di preparazione dei dati è la selezione e l'estrazione delle *features* dal corpus delle recensioni. La scelta delle *features* è un aspetto fondamentale al fine dell'analisi, in quanto, come detto nella sezione 2, per ottenere una buona predizione del fenomeno osservato è fondamentale selezionare le giuste *features* rappresentative dell'oggetto.

Così come per le descrizioni anche le recensioni sono state filtrate in base alla lingua riconosciuta, sono state scartate tutte quelle recensioni che risultavano essere in una lingua diversa dall'italiano o dall'inglese. Questa ha comportato una riduzione dei

dati analizzati, che è andata a sommarsi a quella già ottenuta nel momento in cui sono state scartate alcune descrizioni (sempre filtrate per lingua) e che ha portato alla rimozione automatica delle recensioni associate.

In totale le recensioni perse sono state 4298 sull'insieme iniziale di 23876, di conseguenza le analisi sono state svolte sulle 19578 recensioni rimanenti.

Di seguito vengono riportate le *features* utilizzate per l'analisi.

Presenza keywords: è un valore binario che indica se almeno una delle otto parole chiave è presente all'interno del testo della recensione. Se è presente viene assegnato valore 1, altrimenti valore 0. Questa *feature* sta ad indicare che all'interno della recensione è stato menzionato uno degli aspetti rappresentativi della descrizione dell'applicazione.

Numero keywords: si tratta di un valore numerico che indica il numero di *keywords* contenute all'interno di ogni recensione.

Numero keywords unici: anche questo è un valore numerico che indica il numero di *keywords* uniche contenute all'interno di ogni recensione.

Lunghezza recensione: indica la lunghezza di una recensione espressa in numero di caratteri che la compongono.

Numero punti esclamativi: indica il numero di punti esclamativi presenti in ogni singola recensione. Questa *feature* è stata scelta con la convinzione che gli utenti facciano un uso molto frequente di questi caratteri di punteggiatura nel tentativo di rafforzare il messaggio espresso e che possano avere delle relazioni con il gradimento degli utenti.

Numero punti interrogativi: rappresenta il numero di punti interrogativi nelle recensioni.

Lunghezza serie di esclamativi: si tratta di un valore numerico che indica la lunghezza della serie di punti esclamativi più lunga contenuta nella recensione. Una serie più lunga indica maggiore intensità del concetto espresso, sia esso negativo o positivo.

Lunghezza serie di interrogativi: indica con un valore numerico la lunghezza della serie di punti interrogativi più lunga.

Voto recensione: valore numerico che rappresenta il voto in stelle assegnato alla recensione dall'autore.

Parole positive e parole negative: indicano rispettivamente il numero di parole positive e il numero di parole negative presenti in ogni recensione. Per ottenere tale valore è stato adoperato il dizionario generico di termini polarizzati (vedi sezione 3).

Per ogni parola è stato controllato il valore maggiore tra positivo e negativo per determinare la sua orientazione. Le due *features* con un valore numerico indicano la quantità di termini positivi o negativi che l'utente ha utilizzato per esprimere la sua opinione all'interno della recensione.

Valore di sentimento: questa *feature* è un valore numerico che indica attraverso un unico valore il grado di positività o negatività di sentimento espresso dall'utente nella recensione. Tale valore è stato calcolato sottraendo il valore di negatività di una parola (valore ottenuto tramite il dizionario di termini polarizzati) al suo valore di positività. Se una parola ottiene un valore maggiore di 0 si considera positiva, con grado di positività che è il risultato della sottrazione. Se il risultato è invece minore di 0 si considera negativo con il suo grado associato. Successivamente, per ogni recensione sono stati sommati i valori, così ottenuti, di tutte le parole appartenenti al dizionario che la compongono. All'interno del corpus di recensioni tale valore ricopre l'intervallo che va da -3,60 a 6,08.

Numero aggettivi: rappresenta il numero di aggettivi presenti in ogni recensione.

Numero avverbi: indica il numero di avverbi per ogni recensione.

Numero verbi condizionali: indica il numero di verbi di modo condizionale presenti in una recensione.

Numero verbi futuri: rappresenta il numero di verbi di tempo futuro che sono riportati in ogni recensione.

Presenza congiunzioni avversative: è un valore binario che indica con 0 l'assenza

di congiunzioni coordinative avversative e con 1 la loro presenza. Per ottenere questo valore è stato creato un insieme delle più comuni congiunzioni avversative (ma, però, anzi, tuttavia, eppure, peraltro, piuttosto) e verificato se almeno una di loro fosse presente nella recensione.

-5-

Fase 2: analisi dei dati

La creazione dei corpora e l'estrazione delle *features* sono la prima fase di un processo che si completa con l'analisi dei dati e la successiva interpretazione dei risultati ottenuti.

L'utilizzo di procedure automatizzate per la costruzione dei corpora non permette di avere un completo controllo sul risultato finale. La fase di analisi dei dati ha permesso in primo luogo di comprendere l'effettiva composizione dei corpora oggetto di studio, rispetto a caratteristiche e aspetti necessari a questa indagine.

L'analisi svolta sul corpus delle recensioni ha mostrato una netta maggioranza di recensioni con valutazione positiva (4 o 5) che ricoprono più della metà del totale delle recensioni. Le recensioni con voto 1 sono l'insieme più popoloso dopo quello delle recensioni con voto 5, a discapito delle recensioni con voto 2 o 3 che sono presenti in numero molto ridotto.

Star	Frequenza	Frequenza in percentuale	App gratuite	App a pagamento
1	4366	22%	3592	774
2	1579	11%	1284	295
3	2118	8%	1611	507
4	3739	19%	2690	1049
5	7776	40%	5262	2514

Tabella 2. Dati corpus recensioni.

Come è facile verificare osservando la tabella 2 le recensioni con voto 1 costituiscono il 22% del corpus, le recensioni con voto 2 solo l'8%, quelle con voto 3 l'11%, quelle con voto 4 il 19% e infine l'insieme più grande è quello delle recensioni con voto 5 che occupa il 40% del totale.

L'insieme delle recensioni, nonostante una maggioranza di valutazioni positiva, risulta comunque abbastanza bilanciato, dato che presenta la media di voti di 3,46.

Di conseguenza per quanto riguarda il corpus delle descrizioni il voto medio relativo alle applicazione è un voto che oscilla tra 2.4 e 5, con una media di 4,21.

Si rende necessario sottolineare che le applicazioni studiate sono state selezionate tra le più popolari per ogni categoria, recuperando le prime in classifica, quindi con ogni probabilità questo ha influenzato la differenza di numero tra le recensioni positive e negative nel corpus. Tuttavia si crede che ciò non influisca negativamente sull'indagine svolta, in quanto mira a scoprire relazioni tra aspetti del testo e voto associato.

Inoltre, osservando la tabella 2 si può anche notare che, nonostante le recensioni sono state recuperate in ugual misura da uno stesso numero di applicazione a pagamento e di applicazione gratuite, le applicazioni gratuite risultano maggiormente commentate, infatti solo 5139 recensioni si riferiscono ad applicazioni a pagamento, mentre 14439 recensioni sono relative ad applicazione gratuite, cioè il 74% del corpus. Non sono state riscontrate altre differenze rilevanti tra questi due gruppi di recensioni.

5.1 Domanda RQ1

Nella sezione 2 sono stati posti dei quesiti ai fini di verificare la tesi proposta in questo lavoro. La prima domanda è stata: ci sono aspetti nelle descrizioni di un'applicazione che influenzano l'opinione degli utenti sull'applicazione stessa?

Nel tentativo di rispondere a tale quesito sono state utilizzate le parole chiave estratte dalle descrizioni (vedi sezione 2, 3). Si rende necessario dunque dare una panoramica di come le parole chiave si presentano nel corpus di recensioni e in quale tipologia di

recensioni si ipotizza siano presenti.

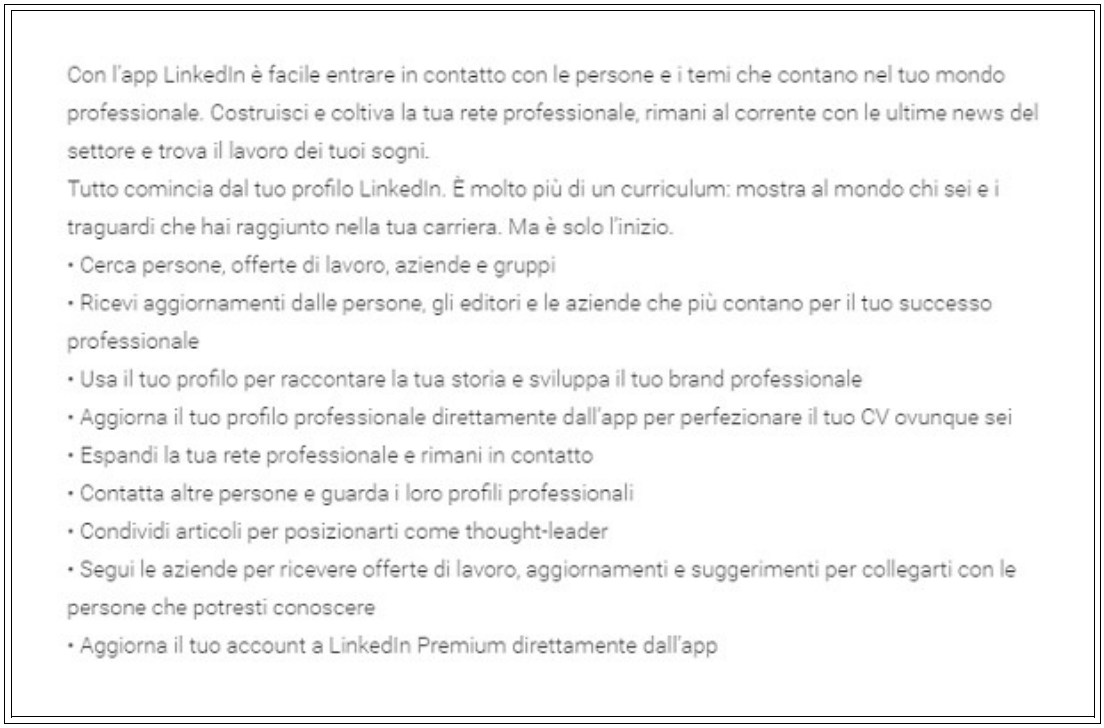

Con l'app LinkedIn è facile entrare in contatto con le persone e i temi che contano nel tuo mondo professionale. Costruisci e coltiva la tua rete professionale, rimani al corrente con le ultime news del settore e trova il lavoro dei tuoi sogni.
Tutto comincia dal tuo profilo LinkedIn. È molto più di un curriculum: mostra al mondo chi sei e i traguardi che hai raggiunto nella tua carriera. Ma è solo l'inizio.
• Cerca persone, offerte di lavoro, aziende e gruppi
• Ricevi aggiornamenti dalle persone, gli editori e le aziende che più contano per il tuo successo professionale
• Usa il tuo profilo per raccontare la tua storia e sviluppa il tuo brand professionale
• Aggiorna il tuo profilo professionale direttamente dall'app per perfezionare il tuo CV ovunque sei
• Espandi la tua rete professionale e rimani in contatto
• Contatta altre persone e guarda i loro profili professionali
• Condividi articoli per posizionarti come thought-leader
• Segui le aziende per ricevere offerte di lavoro, aggiornamenti e suggerimenti per collegarti con le persone che potresti conoscere
• Aggiorna il tuo account a LinkedIn Premium direttamente dall'app

Figura 19. Esempio descrizione applicazione.

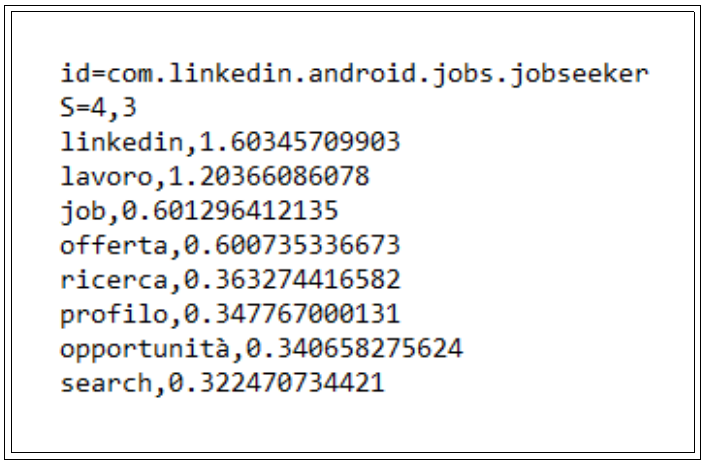


Figura 20. Parole chiave estratte dalla descrizione in figura 19.

Le parole chiave estratte dalla descrizione rappresentano proprietà o funzionalità dell'applicazione. Nell'immagine in figura 19 si può vedere un esempio di

descrizione con relativo insieme di *keywords* e il loro valore TF-IDF (figura 20).

Queste *keywords* si ipotizza che si riscontrino in recensioni scritte con lo scopo di descrivere l'applicazione negli aspetti che la caratterizzano, in questo modo non si può con certezza selezionare o escludere recensioni che propongono nuove funzionalità o che recensiscono bug tecnici dell'applicazione. Ad esempio in una recensione in cui un utente invita gli sviluppatori a inserire nuove funzionalità non è sicuro che vengano nominate funzionalità già esistenti nell'applicazione.

L'idea è che la presenza di parole chiave in una recensione sia indice di un'intenzione da parte dell'utente di recensire l'applicazione, invece che limitarsi ad esprimere un breve giudizio complessivo.

Nelle recensioni prese in esame 6076 presentano almeno una *keyword* e le restanti 13502 non ne presentano alcuna. Nell'insieme di recensioni con voto 1 solo il 28 % delle recensioni presenta almeno una *keyword*, nel gruppo di recensioni con voto 2 le *keyword* sono presenti nel 31% del totale, nelle recensioni con voto 3 nel 30%, in quelle con voto 4 nel 33% e infine nell'insieme di recensioni con voto 5 il 32% delle recensioni presenta almeno una *keyword*. Si nota un lieve aumento della presenza delle *keywords* all'aumentare del voto della recensione, complessivamente nelle recensioni con voto negativo (1 o 2) le *keywords* si presentano nel 29 % del totale mentre nelle recensioni positive (4 o 5) il 32 % delle recensioni contiene almeno una *keyword*.

Queste percentuali relativamente basse sono probabilmente dovute alla scarsa intenzione da parte degli utenti di recensire veramente un'applicazione. Preferiscono piuttosto esprimere un giudizio complessivo su di essa, ad esempio “Fantastica app!”.

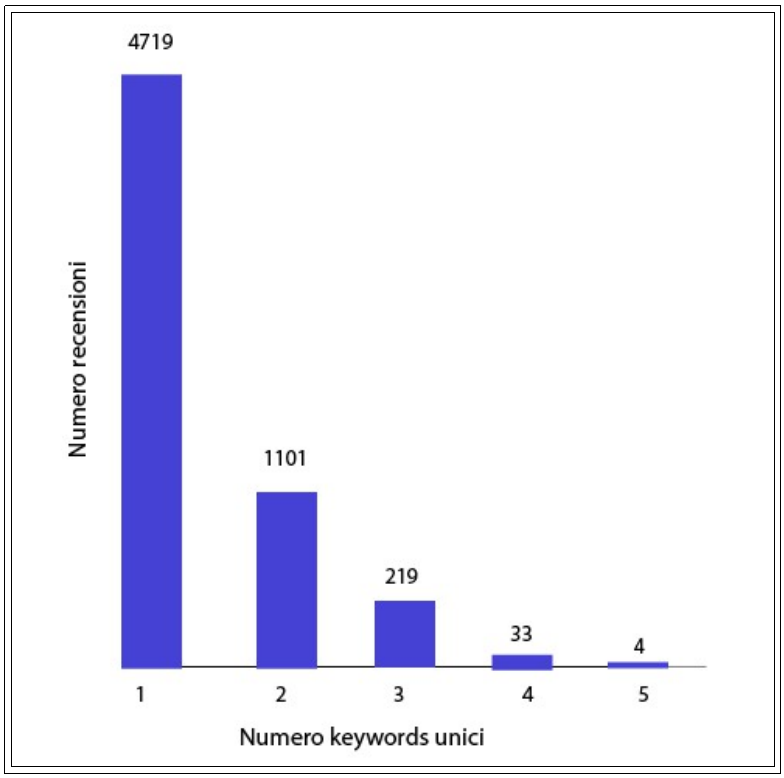


Figura 21. Distribuzione keywords unici.

Inoltre sono pochi gli utenti che richiamano più di un aspetto dell'applicazione nelle recensioni, spesso viene commentata una sola caratteristica alla volta come si può vedere in figura 21, dove vengono prese in considerazione solo le 6076 recensioni in cui appare almeno una *keyword*. Questo può derivare dalla lunghezza media delle recensioni di applicazioni che sono testi generalmente brevi, infatti all'interno del corpus la lunghezza delle recensioni in caratteri è un valore che varia da 22 caratteri (in quanto come detto nella sezione 4 le recensioni più brevi sono state filtrate perché si è ipotizzato fossero poco informative) a 3122, con una lunghezza media di 149 caratteri. Le recensioni di applicazioni sono quindi brevi messaggi molto differenti da altri generi di recensioni che si trovano su internet (come recensioni di film, di hotel, di ristoranti, ecc.).

5.1.1 Ipotesi 1

Le *keywords* appaiono in quasi un terzo delle recensioni totali, di conseguenza è lecito aspettarsi che tali recensioni siano quelle generalmente più lunghe, in quanto l'atto di recensire necessita di un certo numero di caratteri per riuscire a esprimere la propria opinione e motivarla.

Tenendo conto di ciò e sulla base della panoramica sopra esposta si ipotizza che la presenza delle *keywords* debba essere riscontrata nelle recensioni più lunghe e generalmente positive.

Formalizzando quanto espresso:

- *"si ipotizza che la positività del voto espresso dall'utente in una recensione sia in relazione con la presenza delle keywords nelle recensioni con lunghezza maggiore"*.

Per fare tale verifica si è fatto uso di Weka, utilizzando come classificatore M5P .

Le *features* utilizzate sono state la lunghezza della recensione, la presenza delle *keywords* e il voto della recensione.

Il classificatore utilizzato in questo esperimento, come si può vedere in figura 22, ha generato tre regole che dipendono dalla lunghezza delle recensioni.

```
length <= 127.5 : LM1 (10420/98.405%)
length >  127.5 :
|   length <= 233.5 : LM2 (6060/99.038%)
|   length >  233.5 : LM3 (3098/99.309%)

LM num: 1
star =
          -0.0041 * length
          + 0.2229 * presKeywords=True
          + 3.9152

LM num: 2
star =
          -0.0019 * length
          + 0.2925 * presKeywords=True
          + 3.5511

LM num: 3
star =
          0 * length
          + 0.1951 * presKeywords=True
          + 2.986

Number of Rules : 3


Time taken to build model: 1.46 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.1466
Mean absolute error                      1.3948
Root mean squared error                  1.5772
Relative absolute error                 97.4531 %
Root relative squared error             98.9195 %
```

Figura 22. Esperimento ipotesi 1 con M5P.

Si nota che nelle recensioni con una lunghezza minore della media (<=127.5) la presenza delle *keywords* incide in positivo di 0.22 sulla valutazione di una recensione. Per quanto riguarda le recensioni con lunghezza compresa tra 127.5 e

233.5 la presenza delle *keywords* continua a incidere in positivo con un valore di 0.29. Infine a lunghezza superiore a 233.5 si riscontra una diminuzione dell'incidenza da parte della presenza delle *keywords*. Da questo si evince che nelle medie lunghezze queste parole hanno una maggiore incidenza positiva sulla valutazione espressa dall'utente e nelle recensioni con lunghezza maggiore tali *keywords* presentano una riduzione di tale incidenza.

Le cause di tale riduzione possono essere ricondotte a un notevole calo della presenza delle *keywords* nelle recensioni oppure a una distribuzione più bilanciata di esse tra recensioni positive e negative. Per capire da cosa deriva tale fenomeno si è deciso di indagare ulteriormente.

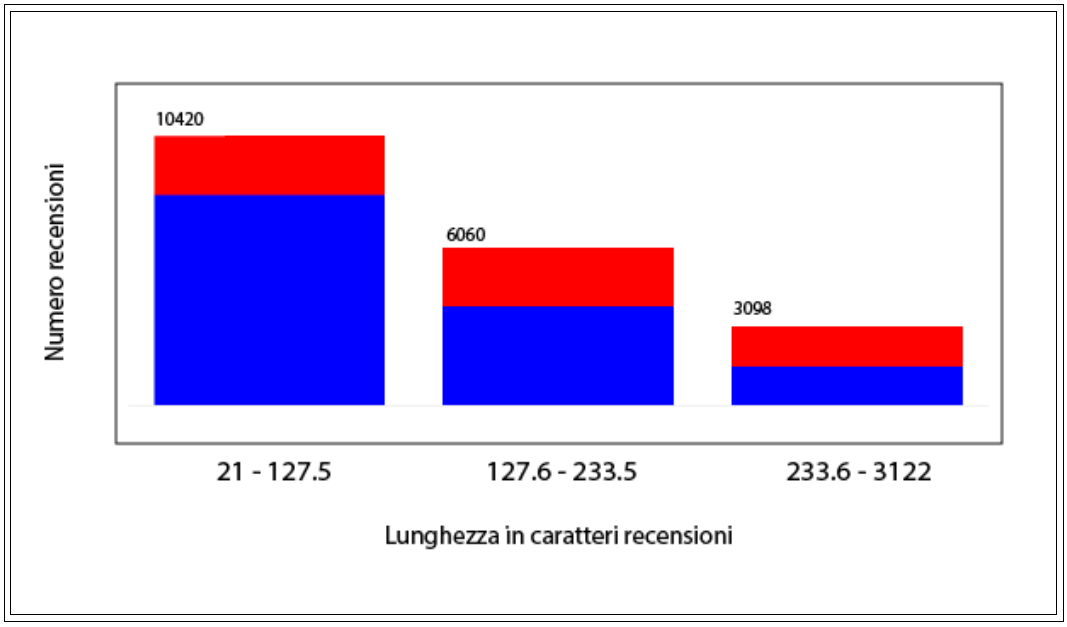


Figura 23. Esperimento ipotesi 1 con M5P.

Il grafico sopra mostrato (figura 23) è stato ottenuto attraverso la rappresentazione grafica dei dati con Weka. L'intero corpus di recensioni è stato suddiviso nelle tre classi di lunghezza risultate dall'albero creato dal classificatore M5P e sono rappresentate dalla tre colonne. Le porzioni di colore blu indicano le recensioni che non presentano *keywords*, mentre le rosse sono quelle che ne presentano almeno una.

Come si può vedere facilmente dal grafico la differenza tra la porzione rossa e quella

blu diminuisce con l'aumentare della lunghezza, e nell'insieme di recensioni a lunghezza più alta addirittura tale differenza si avvicina allo 0, infatti le due porzioni sono circa della stessa dimensione. Ciò dimostra che la riduzione di incidenza delle *keywords* nelle recensioni con lunghezza superiore a 233.5 non dipende da una loro scarsa presenza. Si può dedurre che la causa di tale riduzione sia dovuta a una distribuzione bilanciata della presenza delle *keywords* tra recensioni positive e negative.

5.2 Domanda RQ2

Per rispondere al secondo quesito sono stati analizzati i testi delle recensioni con lo scopo di verificare quale è l'uso effettivo da parte degli utenti della scala di valutazione. L'analisi svolta mira a determinare se recensioni con valutazioni differenti mantengano questa differenza anche nel testo attraverso aspetti linguistici caratterizzanti.

La scala di valori da 1 a 5 presenta due valori per esprimere un'opinione negativa (1 e 2), due valori per esprimere un'opinione positiva (4 e 5) e un valore intermedio (3) di difficile interpretazione. In un primo luogo è stata verificata una possibile differenza che intercorre tra testi che fanno riferimento a valutazioni negative e testi che fanno riferimento a valutazioni positive, successivamente è stata analizzata la coppia che esprime positività e la coppia che esprime negatività, mettendo a confronto tra loro gli elementi che le compongono. Infine si conclude l'indagine confrontando il valore intermedio con i valori ad esso adiacenti (2 e 4) e con i valori estremi (1 e 5).

5.2.1 Ipotesi 1

Se la scala di valutazione proposta negli app store è ben interpretata dagli utenti si può pensare che il valore di sentimento utilizzato in questo lavoro (vedi sezione 4) cambi a seconda del voto espresso nelle recensioni. Quindi si ipotizza che:

- *"all'aumentare del valore di sentimento aumenta il voto in stelle (o viceversa)".*

Per verificare tale ipotesi è stata calcolata la correlazione lineare che intercorre tra il valore di sentimento e il voto, usando il classificatore Linear Regression in Weka. Il valore di correlazione risultante è di 0.46 (figura 24), che permette di confermare l'ipotesi dimostrando che questi due variabili variano insieme con un certa regolarità.

```
Linear Regression Model

star =

      0.8762 * ValoreSentiment +
      2.9822

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.4626
Mean absolute error                      1.2058
Root mean squared error                  1.4135
Relative absolute error                 84.2475 %
Root relative squared error             88.6522 %
Total Number of Instances           19578
```

Figura 24. Esperimento ipotesi 2 con Linear Regression.

Inoltre, suddividendo le recensioni in cinque gruppi in base al voto, si può osservare che il valore medio di sentimento di ogni gruppo aumenta all'aumentare del voto:

1. il gruppo con voto 1 ha valore medio di -0.05;
2. il gruppo con voto 2 ha valore medio di 0.14;
3. il gruppo con voto 3 ha valore medio di 0.37;

4. il gruppo con voto 4 ha valore medio di 0.78;
5. il gruppo con voto 5 ha valore medio di 0.89.

5.2.2 Ipotesi 2

Aggettivi e avverbi sono due categorie grammaticali che in italiano ricoprono la funzione di modificatori semantici. Gli aggettivi sono generalmente usati come modificatori di sostantivi mentre gli avverbi sono usati principalmente come modificatori di un verbo, ma anche di un aggettivo, di un altro avverbio o di un'intera frase. Queste due categorie grammaticali, in particolare gli aggettivi, spesso si presentano in testi descrittivi. L'utilizzo di queste parti del discorso è sicuramente un aspetto interessante nell'ambito di recensioni che sono appunto testi che hanno lo scopo di descrivere un prodotto, un servizio, ecc. Ad esempio la recensione estratta dal corpus "*Consigliata Semplice da utilizzare anche per chi mastica poco l'inglese, scansioni molto buone* " mostra l'intento di descrivere un aspetto di un'applicazione utilizzando un aggettivo e un avverbio.

Sulla base dei risultati dell'ipotesi mostrati nel paragrafo 5.1.1 le recensioni che maggiormente sono utilizzate per descrivere sono quelle positive, quindi si ipotizza che:

- *"la presenza di aggettivi e avverbi nelle recensioni sia relazionata al voto positivo espresso dall'utente"*

In questo caso è stato utilizzato l'albero di regressione M5P per valutare l'incidenza di queste due categorie grammaticale sul voto assegnato alle recensioni.

```
avverbi <= 1.5 :
|   aggettivi <= 1.5 : LM1 (4295/96.621%)
|   aggettivi >  1.5 : LM2 (3885/81.22%)
avverbi >  1.5 :
|   aggettivi <= 1.5 : LM3 (4294/101.709%)
|   aggettivi >  1.5 :
|   |   avverbi <= 3.5 : LM4 (3843/93.373%)
|   |   avverbi >  3.5 : LM5 (3549/96.889%)

LM num: 1
stelle =
    0.5089 * aggettivi
    - 0.6136 * avverbi
    + 3.596

LM num: 2
stelle =
    0.0864 * aggettivi
    - 0.2753 * avverbi
    + 4.0477

LM num: 3
stelle =
    0.3882 * aggettivi
    - 0.131 * avverbi
    + 3.0229

LM num: 4
stelle =
    0.1374 * aggettivi
    - 0.1729 * avverbi
    + 3.6802

LM num: 5
stelle =
    0.1157 * aggettivi
    - 0.1217 * avverbi
    + 3.3617

Number of Rules : 5

Time taken to build model: 0.72 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.3271
Mean absolute error                      1.2979
Root mean squared error                  1.5065
Relative absolute error                 90.7272 %
Root relative squared error             94.4939 %
```

Figura 25. Esperimento ipotesi 2 con M5P.

I risultati in figura 25 mostrano che l'incidenza degli aggettivi come previsto ha valore positivo e quindi indica una tendenza a ritrovare gli aggettivi maggiormente in recensioni positive, al contrario degli avverbi che invece presentano un'incidenza di segno opposto. Questo si verifica soprattutto nella prima regola dove in base alla presenza o assenza di un avverbio o di un aggettivo la recensione può avere voto 4 oppure voto 3.

5.2.3 Ipotesi 3

Nei contesti informali in rete (blog, forum, social network, ecc.) è prassi comune stravolgere la punteggiatura dell'italiano standard cercando di compensare la mancanza di contatto nel tentativo di esprimere uno stato d'animo (M.Tavosanis 2011).

Ad esempio nella recensione "*Crash! Chiede aggiornamenti... ma quali???*" (estratta dal corpus contenete le recensioni) l'utente fa un uso eccessivo di alcuni segni di punteggiatura per riflettere la sua insoddisfazione dovuta a un malfunzionamento dell'applicazione.

Quindi si ipotizza che :

- *"le sequenze di punti interrogativi e punti esclamativi nelle recensioni sia un fattore di valutazione negativa da parte dell'utente".*

Il classificatore M5P ha dato il risultato in figura 26. Osservando tale risultato si nota che i punti esclamativi non incidono, e invece i punti interrogativi hanno una decisa incidenza negativa sulla valutazione espressa nella recensione.

```
LM num: 1
star =
    -0.4126 * lengthInter
    - 0.0412 * lengthExclam
    + 3.5374

Number of Rules : 1

Time taken to build model: 0.32 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.125
Mean absolute error                      1.4099
Root mean squared error                  1.5817
Relative absolute error                 98.5548 %
Root relative squared error             99.2137 %
```

Figura 26. Esperimento ipotesi 3 con M5P.

5.2.4 Ipotesi 4

I voti 4 e 5 compongono la coppia di valori positivi nella scala di valori da 1 a 5. Il voto 5 rappresenta il voto massimo quindi dovrebbe essere utilizzato solo nei casi in cui l'utente è completamente soddisfatto. Seguendo tale ragionamento il voto 4 dovrebbe essere utilizzato da utenti soddisfatti ma che riportano uno o più aspetti di minore importanza che non li ha pienamente convinti.

Le recensioni con voto 4 dovrebbero presentare al loro interno aspetti linguistici che indicano un cambio di orientamento (positivo o negativo) all'interno del discorso.

Si è deciso dunque di aggiungere all'analisi la presenza di congiunzioni avversative in quanto sono utilizzate in italiano per indicare parole o proposizioni in contrasto tra loro.

L'ipotesi che si vuole verificare è:

- "*le recensioni con voto 4 e con voto 5 dovrebbero presentare differenze linguistiche all'interno dei loro testi riconducibile a un diverso grado di soddisfazione*".

Il primo esperimento per verificare tale ipotesi coinvolge appunto la presenza di congiunzioni avversative oltre agli aspetti che differenziano le recensioni per positività e negatività.

```
LM num: 1
star =
    0.0067 * aggettivi
    - 0.0171 * avverbi
    - 0.1863 * presenzAvversative
    - 0.0484 * lengthInter
    - 0.0531 * numNegative
    + 0.0206 * numPositive
    + 4.7115

Number of Rules : 1

Time taken to build model: 0.79 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.2259
Mean absolute error                      0.4145
Root mean squared error                  0.4554
Relative absolute error                 94.819  %
Root relative squared error             97.4116 %
```

Figura 27. Esperimento ipotesi 4 con M5P.

Osservando la figura 27 si può notare che tutti gli aspetti linguistici indicatori di negatività o positività non risultano rilevanti ai fini di differenziare i due gruppi di

recensioni. L'unico aspetto che permette di riscontrare delle differenza è la presenza o meno all'interno del testo delle recensioni di congiunzioni avversative, le quali incidono negativamente quindi verso le recensioni con voto 4.

Le recensioni con voto 4 dunque generalmente si distinguono da quelle con voto 5 perché presentano riferimenti ad aspetti minori che gli utenti non hanno pienamente gradito. Dovrebbe essere lecito quindi aspettarsi che questi utenti rispetto a quelli che assegnano una valutazione massima siano più portati a fornire consigli e suggerimenti agli sviluppatori.

Nel secondo esperimento si sono volute analizzare le differenze tra le recensioni con voto 5 e con voto 4 attraverso lo studio di verbi di tempo futuro e verbi di modo condizionale, utilizzati solitamente quando si vuole dare un consiglio.

```
LM num: 1
star =
     0.0117 * aggettivi
     - 0.0123 * avverbi
     - 0.177 * presenzAvversative
     - 0.1232 * Numfuture
     - 0.1624 * Numcond
     - 0.0438 * lengthInter
     - 0.0532 * numNegative
     + 0.0192 * numPositive
     + 4.7147

Number of Rules : 1

Time taken to build model: 0.87 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.2603
Mean absolute error                      0.4066
Root mean squared error                  0.4514
Relative absolute error                 93.025  %
Root relative squared error             96.5543 %
```

Figura 28. Esperimento ipotesi 4 con M5P.

I risultati che si possono osservare in figura 28 confermano le aspettative, dimostrando che vi è una tendenza nelle recensioni con valutazione 4 ad utilizzare verbi in tempo futuro e verbi di modo condizionale rispetto alle recensioni con valutazione 5.

L'ipotesi posta è stata provata in quanto i risultati degli esperimenti hanno mostrato che esistono differenze linguistiche, seppur lievi, tra le recensioni con voto 4 e le recensioni con voto 5.

5.2.5 Ipotesi 5

I valori più bassi della scala (1 e 2) si ipotizza che presentino differenze analoghe ai valori 4 e 5 nei testi delle recensioni. Come nelle recensioni con voto 4 anche in quelle con voto 2 dovrebbe avvenire un cambio di orientamento nel testo verso positività o negatività.

Formalizzando tale ipotesi:

- *"si ipotizza che le recensioni con voto 2 presentano aspetti linguistici differenti dalle recensioni con voto 1, dovuti a un differente grado di insoddisfazione".*

É stato svolto l'esperimento utilizzando il classificatore M5P e i risultati sono mostrati in figura 29.

```
LM num: 1
star =
    0.0123 * aggettivi
    - 0.0063 * avverbi
    + 0.0501 * presenzAvversative
    - 0.0188 * numNegative
    + 0.0184 * numPositive
    + 1.2494

Number of Rules : 1

Time taken to build model: 0.33 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.0913
Mean absolute error                      0.386
Root mean squared error                  0.4396
Relative absolute error                 99.0156 %
Root relative squared error             99.555  %
```

Figura 29. Esperimento ipotesi 5 con M5P.

A differenza di quanto riscontrato nelle recensioni con voto 4 e 5, osservando i risultati si nota che le recensioni con voto 1 e 2 non presentano nessuna differenza significativa per quanto riguarda gli aspetti linguistici presi in esame.

Come si può dedurre i risultati non confermano l'ipotesi posta.

5.2.6 Ipotesi 6

L'ultimo valore della scala da analizzare è il voto intermedio (3), in una scala di gradimento con valori da 1 a 5 il voto 3 dovrebbe rappresentare il voto neutro.

Nella situazione in cui a un utente viene posto un questionario l'utente si trova obbligato a dover esprimere il suo giudizio a ogni affermazione posta. Alcune affermazioni potrebbero non suscitare l'interesse dell'utente, il quale ha la possibilità

di esprimere la sua indifferenza con il voto neutro.

Nel contesto degli app store all'utente è data la possibilità di valutare e recensire un'applicazione ma non è obbligato a farlo. È dunque difficile credere che un utente, di sua spontanea iniziativa, valuti un'applicazione per esprimere indifferenza.

Immaginarsi a priori quale possa essere l'utilizzo effettivo che viene fatto dagli utenti che scrivono una recensione con valutazione 3 è il compito più arduo. Si ipotizza dunque che:

- "*l'insieme di recensioni con valutazione 3 presenta caratteristiche differenti rispetto alla coppia di recensioni positive e a quella di recensioni negative.*"

I primi due esperimenti si focalizzano sull'analisi delle differenze che ci sono tra il 3 e il 2 e tra il 3 e il 4, utilizzando gli aspetti linguistici indicatori di positività o negatività visti fin'ora.

```
LM num: 1
star =
    -0.0246 * avverbi
    + 0.103 * presenzAvversative
    + 0.0963 * Numfuture
    + 0.1035 * Numcond
    - 0.033 * numNegative
    + 0.0323 * numPositive
    + 2.5773

Number of Rules : 1

Time taken to build model: 0.34 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.1902
Mean absolute error                      0.4705
Root mean squared error                  0.4856
Relative absolute error                 96.1448 %
Root relative squared error             98.1482 %
```

Figura 30. Esperimento ipotesi 6 con M5P.

```
LM num: 1
star =
    -0.0313 * avverbi
    - 0.0855 * presenzAvversative
    - 0.0682 * lengthInter
    - 0.0436 * numNegative
    + 0.0737 * numPositive
    + 3.6398

Number of Rules : 1

Time taken to build model: 0.61 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.2827
Mean absolute error                      0.4256
Root mean squared error                  0.4609
Relative absolute error                 92.177  %
Root relative squared error             95.9183 %
Total Number of Instances             5922
```

Figura 31. Esperimento ipotesi 6 con M5P.

Nella figura numero 30 si nota che tra le recensioni con voto 2 e le recensioni con voto 3 c'è una lieve incidenza in positivo di quelli aspetti (presenze delle congiunzioni avversative e numero di verbi con modo condizionale) che caratterizzano le differenze tra i testi delle recensioni con voto positivo.

In figura numero 31 vengono invece riportati i risultati dell'analisi delle differenze del gruppo di recensioni con voto 3 e con voto 4. Non è stata riscontrata nessuna incidenza rilevante che possa permettere di distinguere questi due insiemi di recensioni.

Dati i risultati appena esposti si è voluto approfondire l'analisi indagando nelle differenze che potrebbero esistere tra le recensioni con voto 3 e le recensioni valutate con i valori estremi della scala.

```
presenzAvversative <= 0.5 : LM1 (4852/93.473%)
presenzAvversative >  0.5 : LM2 (1704/100.792%)

LM num: 1
star =
    0.0415 * aggettivi
    - 0.0526 * avverbi
    + 0.001 * presenzAvversative
    + 0.0002 * Numfuture
    + 0.384 * Numcond
    - 0.0001 * lengthInter
    - 0.1016 * numNegative
    + 0.0872 * numPositive
    + 1.6111

LM num: 2
star =
    0.0003 * aggettivi
    - 0.0676 * avverbi
    + 0.0027 * presenzAvversative
    + 0.0006 * Numfuture
    + 0.1811 * Numcond
    - 0.1154 * lengthInter
    - 0.0895 * numNegative
    + 0.1413 * numPositive
    + 2.0232

Number of Rules : 2

Time taken to build model: 0.5 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.284
Mean absolute error                      0.8078
Root mean squared error                  0.8994
Relative absolute error                 91.8051 %
Root relative squared error             95.8636 %
```

Figura 32. Esperimento ipotesi 6 con M5P.

```
numNegative <= 0.5 :
|   numPositive <= 1.5 : LM1 (2905/97.477%)
|   numPositive >  1.5 : LM2 (3779/64.981%)
numNegative >  0.5 : LM3 (3393/106.714%)

LM num: 1
star =
    0.0001 * aggettivi
    - 0.1223 * avverbi
    - 0.414 * presenzAvversative
    - 0.001 * Numfuture
    - 0.242 * Numcond
    - 0.2074 * lengthInter
    - 0.0003 * numNegative
    + 0.2646 * numPositive
    + 4.582

LM num: 2
star =
    0.0124 * aggettivi
    - 0.05 * avverbi
    - 0.3294 * presenzAvversative
    - 0.0008 * Numfuture
    - 0.1554 * Numcond
    - 0.2005 * lengthInter
    - 0.0003 * numNegative
    + 0.0554 * numPositive
    + 4.788

LM num: 3
star =
    0.0001 * aggettivi
    - 0.0508 * avverbi
    - 0.4053 * presenzAvversative
    - 0.1538 * Numfuture
    - 0.1903 * Numcond
    - 0.1331 * lengthInter
    - 0.1129 * numNegative
    + 0.1367 * numPositive
    + 4.4977

Number of Rules : 3

Time taken to build model: 0.65 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.4225
Mean absolute error                      0.5557
Root mean squared error                  0.7415
Relative absolute error                 83.0142 %
Root relative squared error             90.6265 %
```

Figura 33. Esperimento ipotesi 6 con M5P.

I risultati in figura 32 mostrano che le differenze tra le recensioni con voto 1 e le recensioni con voto 3 sono varie anche se non particolarmente incisive. Ciò che fa la differenza è il numero di verbi condizionali che caratterizzano di più le recensioni con voto 3, l'abuso di punti interrogativi che caratterizza le recensioni con voto 1 e la maggior presenza di parole con accezione negativa e positiva rispecchia la scala di voto.

In figura 33 i risultati dell'analisi tra recensioni con voto 3 e con voto 5 mostrano che le differenze tra questi due insiemi sono ancora più numerose e con incidenza maggiore. Le recensioni con voto 3 non presentano una differenza rilevante per numero di parole negative rispetto a quelle con voto 5, le quali però si differenziano per la quantità di parole positive all'interno del loro testo. Verbi al futuro, verbi condizionali e la presenza di congiunzioni avversative definiscono un'incidenza rilevante che caratterizza le recensioni con voto 3, ed anche l'abuso di punti interrogativi, aspetto che indica negatività, detiene un'incidenza significativa per la riduzione del voto delle recensioni.

L'ipotesi affermava l'esistenza di differenze che fossero in grado di distinguere le recensioni con voto intermedio dalle altre. Tale ipotesi è confermata in parte, si notano differenze significative nel confronto tra le recensioni con voto 3 e quelle con voto 5 e 1, mentre queste differenze affievoliscono nel confronto con recensioni con voto 2, fino a sparire rispetto alle recensioni con voto 4.

-6-

Discussione

6.1 Interpretazione risultati RQ1

“Ci sono aspetti nelle descrizioni di un'applicazione che influenzano l'opinione degli utenti sull'applicazione stessa?”

Le *keywords*, come detto più volte, rappresentano proprietà e funzionalità offerte dall'applicazione e menzionate nella sua descrizione.

L'uso di queste parole in una recensione è rappresentativo dell'esperienza vissuta dall'utente nell'utilizzare l'applicazione e della sua opinione riguardo il servizio offerto.

I risultati hanno restituito un esito inaspettato rispetto all'ipotesi posta. Infatti essi mostrano che la presenza delle *keywords* è associata a una recensione positiva a lunghezze basse e medio basse, mentre questa tendenza svanisce a lunghezze superiori alla media.

Si può concludere che l'utente nelle recensioni con lunghezze più basse tende a esprimere la sua esperienza principalmente se tale esperienza è positiva, le recensioni negative in questi casi probabilmente sono usate o per esprimere la propria insoddisfazione o per segnalare bug tecnici. Si può affermare dunque che quando le recensioni rientrano in questa lunghezza in genere per lo sviluppatore può risultare più utile affidarsi a recensioni positive ed evitare recensioni che con poco probabilità danno informazioni di questo genere.

Per quanto riguarda recensioni con lunghezza superiore alla media è facile aspettarsi che contengano maggiori informazioni rispetto alle altre. Ma dai risultati ottenuti è interessante notare che le *keywords* tendono a perdere il loro indice di positività all'aumentare della lunghezza, e che le recensioni che presentano *keywords*

aumentano sempre all'aumentare della lunghezza. Questo gruppo di recensioni può dunque essere considerato quel tipo di recensione in cui l'utente si sofferma a recensire l'applicazione nel senso stretto del termine, esprimendo la sua opinione e dando una valutazione a seconda della sua esperienza che può risultare sia positiva sia negativa in egual misura.

Tuttavia i dati mostrano anche che questo gruppo di recensioni si presenta in numero molto minore rispetto agli altri; si può quindi affermare che queste recensioni siano un'utile fonte di informazione per gli sviluppatori ma che ricorrano con una frequenza molto bassa. Per conoscere quello che gli utenti pensano riguardo al servizio offerto dall'applicazione, evitando recensioni che poco si soffermano a descrivere tale servizio, è però possibile andare a ricercare queste informazioni anche nelle recensioni a lunghezza media con valutazione positiva.

6.2 Interpretazione risultati RQ2

"Qual è l'uso effettivo che fanno gli utenti della scala di valori quando assegnano una valutazione?"

Gli esperimenti effettuati per rispondere a questa domanda hanno mostrato che gli utenti tendono a non fare distinzione tra il voto 1 e il voto 2, al contrario il voto 4 e il voto 5 presentano un diverso grado di positività, che è seguito dal voto 3 che sembra essere una valutazione che, anche se poco utilizzata, esprime apprezzamento degli utenti con accezione più verso la positività.

Dall'osservazione dei risultati dell'ipotesi 1 si nota una scala delle medie dei valori di sentimento riconducibile ai 5 voti della scala di valutazione. Tale iniziale risultato mostra che tra ogni voto vi è una sottile differenza nel grado di positività o negatività espressa, ma da solo questo risultato non permette di capire a fondo qual è l'uso effettivo della scala di valutazione da parte degli utenti.

Le recensioni con voto negativo hanno mostrato caratteristiche distintive di negatività che permettono di riconoscerle dalle altre. Gli aspetti indicatori di

negatività che sono stati individuati sono: il numero di avverbi, l'uso spropositato di punti interrogativi e ovviamente anche un'alta presenza di parole con accezione negativa.

Riguardo alle recensioni positive si può dire che anch'esse possiedono aspetti caratterizzanti che indicano positività: numero di parole positive, numero di aggettivi, e solo in alcune condizioni, i verbi condizionali, i verbi futuri e la presenza di congiunzioni avversative.

L'abuso di punteggiatura sul Web, in contesti non formali, spesso è usato dagli utenti come mezzo per esprimere il proprio stato d'animo. I punti interrogativi, usati nell'italiano standard per esprimere una domanda, vengono qui usati anche per esprimere indignazione e insoddisfazione. Per far trasparire tale sentimento sono utilizzati in sequenze sempre più lunghe, maggiore è l'intenzione da parte dell'utente di mettere in evidenza il proprio sentimento.

I punti esclamativi nell'italiano standard servono per esprimere l'intonazione di un enunciato. Sono associati all'emotività dello scrivente e ciò li rende adatti, in contesti come le recensioni di applicazioni, a rappresentare sentimenti ed emozioni.

Gli utenti utilizzano questo tipo di punteggiatura in sequenza con l'intenzione di rafforzare un sentimento o un'opinione espressa. A differenza dei punti interrogativi i punti esclamativi però sono stati riscontrati sia in contesti positivi che in contesti negativi.

L'atto di descrivere, rappresentato dalla presenza di aggettivi nelle recensioni, è spesso legato ad un'opinione positiva dell'utente, questo è dimostrato dal fatto che gli utenti ne fanno uso principalmente in contesti positivi. Questo aspetto si ricollega alla presenza delle *keywords* indagate nella domanda RQ1 che ha messo in luce una preferenza degli utenti di descrivere un'applicazione solo se è stata apprezzata.

Inaspettatamente, rispetto a quanto detto sopra, gli avverbi invece indicano una forte tendenza negativa, in quanto riscontrati più che altro nelle recensioni con valutazione 1 e 2. Questo fenomeno è facilmente spiegato dall'alta frequenza dell'avverbio di negazione “non” che ovviamente si presenta in numero elevato nelle recensioni negative (un esempio estratto dal corpus è: “*Non funziona!!!! Da qualche giorno*

non funziona più!").

Se può sorprendere l'uso della punteggiatura in relazione alla negatività o alla positività, non stupisce invece che nelle recensioni con voto 1 e 2 vi sia un'elevata presenza di parole con accezione negativa. Questa presenza però non mostra in media differenze tra queste recensioni. Allo stesso modo nessuno degli aspetti negativi individuati ha mostrato differenze tra loro.

Ciò che i risultati hanno evidenziato è che gli utenti tendono a utilizzare questi due voti come se fossero lo stesso, inoltre, come si può notare dal corpus, raramente ricorrono al voto 2 per esprimere la loro valutazione. Gli utenti, dunque, sembrano non sentire la necessità di più voti quando devono esprimere un'opinione negativa circa un'applicazione.

Probabilmente se un'applicazione non svolge la sua funzione o presenta errori che ne compromettono il suo utilizzo, l'utente sente solo il bisogno di manifestare il suo disappunto.

La coppia di valori positivi (4 e 5), opposti nella scala ai valori 1 e 2, manifestano, al contrario di quest'ultimi, differenze significative. Entrambi i voti positivi sono utilizzati per esprimere stesso grado di positività, infatti il numero di parole positive e il numero di parole negative, mediamente utilizzati, non mostra differenze tra i due gruppi di recensioni con voto 4 e 5. Nonostante ciò sono state rilevate differenze che derivano dalle funzionalità che i due voti assumono per gli utenti.

Nella situazione in cui un utente apprezza nella sua totalità un'applicazione è presumibile che nella sua valutazione assegni il voto massimo della scala; invece nella situazione in cui un utente apprezza un'applicazione ma allo stesso tempo pensa che a quest'ultima possano essere apportati miglioramenti, è presumibile che egli assegni voto 4.

La presenza di congiunzioni avversative, in italiano standard, segnala all'interno di un enunciato un cambio di orientamento. Nel caso delle recensioni di applicazioni se inizialmente c'è un orientamento positivo cambierà verso negativo e viceversa (ad esempio "*Amway App Ottima ma manca il carrello...*").

Le congiunzioni avversative sono un primo aspetto che distingue le recensioni con voto 4 da quelle con voto 5, il che indica che mediamente in queste recensioni gli utenti tendono a costruire il testo mettendo aspetti a contrasto. La riprova che questa struttura testuale viene utilizzata per esprimere suggerimenti è data dall'utilizzo maggiore di verbi condizionali e di verbi al futuro, che contraddistinguono le recensioni con voto 4.

L'esistenza delle differenze tra le recensioni con voto 4 e le recensioni con voto 5 mostra che gli utenti effettivamente utilizzano il voto 4 e il voto 5 come due voti distinti, il che evidenzia una necessità da parte degli utenti di poter scegliere tra due sfumature di positività.

Per quanto riguarda il voto 3 si può dire che è il voto meno caratterizzato della scala, in quanto le recensioni con tale valutazione presentano aspetti simili a quelle con valutazione 2 e anche a quelle con valutazione 4.

Rispetto alle recensioni con voto 2 quelle con voto 3 si distinguono per maggiore presenza di congiunzioni avversative e verbi condizionali. Questo mostra che nelle recensioni con voto 3 l'utente mantiene l'interesse a fornire consigli e suggerimenti agli sviluppatori, aspetto che nelle recensioni con voto 2 è poco presente.

Al confronto con recensioni con voto 1 la differenza media aumenta, mettendo in luce un grado maggiore di positività a favore delle recensioni con voto 3 e un grado di negatività verso le recensioni con voto 1.

Le recensioni con voto 4 risultano ancora più simili alle recensioni con voto 3 rispetto a quelle con voto 2. Le differenze tra il 3 e il 4 si notano nel loro rapporto con le recensioni con voto 5, in cui si nota una maggiore differenza tra il 3 e il 5.

Tra le recensioni con voto 3 e quelle con voto 5 tutti gli aspetti studiati influiscono seppur lievemente. Le recensioni con voto 3 rappresentano quasi la soglia tra negatività e positività, anche se con una maggiore tendenza verso il positivo, data anche la loro similarità con le recensioni con voto 4.

Il voto 3 è sicuramente il voto più difficilmente interpretato dagli utenti, che tendono ad utilizzarlo o come un voto positivo o come un voto negativo. Nonostante ciò la

sua funzione pare molto simile a quella svolta dal voto 4, usato maggiormente dagli utenti anche con l'intento di dare consigli e suggerimenti. Inoltre è significativa la scarsa presenza di recensioni con voto 3 all'interno del corpus, di poco superiore alle recensioni con voto 2.

Per concludere, si può dire che come era prevedibile i voti agli estremi della scala sono i più usati, essendo i più facili da interpretare. Dei voti intermedi solo il voto 4 è usato con frequenza con una funzione che accomuna l'interpretazione della maggior parte degli utenti; il voto 2, oltre a essere poco usato, viene utilizzato allo stesso modo del voto 1, mentre il voto 3 tende a ricoprire la funzione del voto 4 con un minor grado di positività, anche se comunque raramente usato.

-7-
Conclusioni

Le recensioni prodotte dagli utenti negli app store sono uno strumento utile sia per gli utenti che hanno la possibilità di esprimere la loro opinione, di chiedere consigli, di segnalare malfunzionamenti o bug tecnici, sia per gli sviluppatori che possono fare buon uso di tutti questi suggerimenti, per rendere il proprio prodotto migliore adattandolo alle esigenze del consumatore.

Sebbene le descrizioni di applicazioni siano il primo strumento con cui lo sviluppatore comunica con gli utenti questi ultimi spesso non si soffermano a commentare aspetti presenti in essa, infatti nel corpus creato per questo lavoro solo un terzo di recensioni si rifanno al testo della descrizione. Questo in parte può essere dovuto ai testi delle descrizioni che riportano in maniera generale le funzionalità offerte dall'applicazione, mentre gli utenti tendono a focalizzarsi su aspetti specifici, come aspetti tecnici, ad esempio malfunzionamenti, o sotto task delle funzionalità principali.

Nonostante le recensioni contenenti *keywords* non siano l'insieme di recensioni più consistente si è potuto notare che ciò che viene descritto nel testo delle descrizioni è legato sia alla lunghezza delle recensioni sia alla valutazione positiva espressa dall'utente. Questo doppio legame, considerata la varietà di tipologie di recensioni che gli utenti forniscono, può aiutare a semplificare la ricerca di informazione all'interno dell'insieme delle recensioni di un'applicazione.

Si crede interessante approfondire maggiormente l'influenza delle descrizioni sull'opinione degli utenti studiando altre parti degli aspetti che le compongono o anche la totalità, oppure proseguire ad analizzare il testo attraverso un approccio più funzionale rispetto a quello usato in questo studio.

In seguito si è voluto analizzare la scala di valutazione e l'utilizzo che gli utenti fanno di essa, con lo scopo di capire se gli utenti nell'atto di valutare trovino lo strumento adatto per soddisfare le loro necessità. Si è osservato che gli utenti tendono ad usare i

due voti negativi (1 e 2) allo stesso modo e raramente fanno ricorso al voto 2. Il voto 3, essendo il voto centrale, risulta un voto ambiguo e come era facile prevedere le recensioni votate 3 si assomigliano, negli aspetti considerati, a recensioni valutate 2 o 4.

Nelle recensioni con voto 3 si è osservato che esse presentano spesso consigli e suggerimenti, aspetto che caratterizza maggiormente le recensioni con voto 4. Questo, unito allo scarso uso che gli utenti fanno di tale voto, mostra che il 3 è sicuramente un voto che gli utenti mal volentieri usano e quando lo fanno, lo utilizzano per esprimere consigli con un giudizio più tendente alla positività che alla negatività.

Le recensioni con voto 4 e 5 risultano entrambe positive ma con funzionalità diverse: il voto 5 serve per esprimere totale apprezzamento, il voto 4 invece viene utilizzato dagli utenti per assegnare un voto sempre positivo ma con l'aggiunta nel testo della recensione di qualche consiglio o proposta. Tale netta distinzione di utilizzo ci fa credere che, insieme al voto 1, siano i voti che la maggior parte degli utenti interpreti allo stesso modo e dunque che tali voti siano più facili da capire e utilizzare. Infatti, l'utilizzo di questi 3 voti ricopre l'81% del totale delle recensioni nel copus analizzato.

Si conclude che la scala di valori riesce a soddisfare le esigenze degli utenti ma questi ultimi risulterebbero essere ancor più soddisfatti se avessero una scala più semplice per poter esprimersi. Quando la loro opinione è negativa essa può essere espressa con un unico voto, come spesso mostrano di fare. Quando la loro opinione è positiva essi mostrano una tendenza a valutare e commentare con più attenzione un'applicazione e quindi sentono la necessità di avere a dispoizione più voti per esprimere la loro opinione.

Una scala a 3 valori sembrerebbe una buona soluzione: un unico valore negativo (1) che raggrupperebbe tutte le recensioni degli attuali voti negativi, e due valori positivi (2 e 3) che ricoprirebbero il ruolo dei tre voti più alti della scala o sicuramente del 4 e del 5. Inoltre per rendere l'utente consapevole della distinzione di valutazioni, e non creare confusione nell'uso del voto intermedio, come succede attualmente con il voto 3, sarebbe anche interessante prendere in considerazione di rappresentare la scala in

stelle con l'aggiunta di colori che suddividano graficamente il voto negativo dai due voti positivi. Una scala divisa per colori che distinguono le valutazioni positive da quelle negative renderebbe l'utente più consapevole dei voti da associare al testo contenete la propria opinione, e si potrebbe aiutare ulteriormente l'utente attraverso l'uso di una scala ridotta più semplice da interpretare.

Per concludere si propone, dunque, l'uso di una scala con tre valutazioni (1,2,3) e due colori per vedere come gli utenti si approccerebbero nel caso in cui dovessero esprimere la propria opinione associandola a questa nuova scala proposta e sarebbe interessante studiare se tali recensioni rispecchiano le aspettative.

Bibliografia

1. Attardi G., Dei Rossi S., Simi M, *The Tanl Pipeline*. In Proceedings LREC 2006.

2. Basile V. e Nissim M., *Sentiment analysis on Italian tweets.* In 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2013.

3. Bird S., Klein E. e Loper E., *Natural language processing with Python*. A cura di O'Reilly Media, 2009.

4. Blumberg R., Atre S., *The Problem with Unstructured Data,* 2003.

   http://www.information-management.com/issues/20030201/6287-1.html

5. Bouckaert Remco R., Frank E., Hall M., Kirkby R., Reutemann P., Seewald A., Scuse D., *WEKA Manual for Version 3-7-12*, 2014.

6. Chiari I., *Introduzione alla linguistica computazionale*. A cura di Editori Laterza, 2007.

7. Cord M., Cunningham P. e Delany S., *Machine Learning Techniques for Multimedia*. A cura di Springer, pp. 21-49, 2008.

8. Corino E., *Educazione Linguistica, Language Education*. A cura di Ca' Foscari, pp.234, 2014.

9. Dulli S., Polpettini P, e Trotta M., *Text mining: teoria e applicazioni.* A cura di Franco Angeli, 2004.

10. Esuli A. e Sebastiani F., *SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining.* In Proceedings of Language Resources and Evaluation (LREC), 2006.

11. Esuli A. e Sebastiani F., *Determining term subjectivity and term orientation for opinion mining*. In Proceedings of EACL-06, 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, IT. Forthcoming, 2006.

12. Grimes S., *Unstructured data and the 80 percent rule, 2008.*

    https://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/

13. Guzman E. e Maalej W., *How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews*. In IEEE International Requirements Engineering Conference, 2014.

14. Harman M., Jia Y. e Zhang Y., *App Store Mining and Analysis: MSR for App Stores*. In 9th IEEE Working Conference on Mining Software Repositories, 2012.

15. Hastie T., Tibshirani R. e Friedman J., *The Elements of Statistical Learning*. A cura di Springer, pp. 485, 2008.

16. Kilgarriff A. e Grefenstette G., *Introduction to the Special Issue on the Web as Corpus*. In Computational Linguistics n. 23, 2003.

17. Lenci A., Montemagni S. e Pirelli V., *Testo e Computer – Elementi di linguistica computazionale*. A cura di Carocci editore, 2005.

18. Lenci A. e Calzolari N., *LINGUISTICA COMPUTAZIONALE STRUMENTI E RISORSE PER IL TRATTAMENTO AUTOMATICO DELLA LINGUA.* In Mondo Digitale, 2004.

19. Liu B., *Sentiment Analysis and Subjectivity.* In Handbook of Natural Language Processing, Second Edition, 2010.

20. Mitchell T., *Machine Learning*. A cura di McGraw-Hill Companies, pp.3, 1997.

21. Moreno-Ortiz A. e Hernández C., *Lexicon-Based Sentiment Analysis of Twitter Messages in Spanish.* In Procesamiento del Lenguaje Natural, Revista n. 50 marzo de 2013.

22. Pagano D. e Maalej W., *User Feedback in the AppStore: An Empirical Study.* In IEEE International Requirements Engineering Conference, 2013.

23. Pang B. e Lee L., *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*. In Proceedings of the ACL, 2005.

24. Ross J., *Quinlan: Learning with Continuous Classes*. In 5th Australian Joint Conference on Artificial Intelligence, Singapore, pp. 343-348, 1992.

25. Tavosanis M., *L'italiano del web*. A cura di Carocci editore, 2011.

26. Wang ,Y., Witten I., *Inducing Model Trees for Continuous Classes*. In Poster papers of the 9th European Conference on Machine Learning, 1997

27. Wilson R. e Keil F., *The MIT Encyclopedia of the Cognitive Sciences,* 1999.